

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**

# **DIPLOMOVÁ PRÁCE**

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra kybernetiky a biomedicínského inženýrství**

**Vestavné diagnostické rozhraní elektromobilu**  
**Built-in Electric Car Diagnostic Interface**

## Zadání diplomové práce

Student: **Bc. Zdeněk Fajkus**  
Studijní program: N2649 Elektrotechnika  
Studijní obor: 2601T004 Měřicí a řídicí technika  
Téma: **Vestavné diagnostické rozhraní elektromobilu**  
**Built-in Electric Car Diagnostic Interface**

Zásady pro vypracování:

1. Seznámit se s problematikou elektromobilu.
2. Seznámit se se sběrnici CAN.
3. Seznámit se s protokolem CANOpen.
4. Navrhnout řešení pro diagnostické rozhraní elektromobilu.
5. Realizovat diagnostický modul.
6. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

1. HUSAIN, I. *Electric and Hybrid Vehicles: Design Fundamentals.*, 2nd Edition. CRC Press, 2010. 288 s. ISBN-13: 9781439811757.
2. VOSS, W. *A Comprehensive Guide to Controller Area Network.* Copperhill Media Corporation, 2005. 164 s. ISBN-13: 978-0976511601.
3. SANTIN, J.J. et al. *The world's most fuel efficient vehicle : design and development of PAC-Car II.* Zürich : Verlag der Fachvereine Hochschulverlag AG an der ETH Zurich, c2007. 352 s. ISBN 978-3-7281-3134-8 (brož.).
4. LARMINIE, J. - LOWRY, J. *Electric vehicle technology explained.* Chichester(UK): John Wiley and Sons, 2003. 296 s. ISBN 0-470-85163-5 (váz.).

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. David Vala**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



doc. Ing. Jiří Koziorek, Ph.D.  
vedoucí katedry

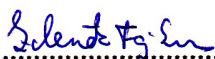


prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

### Prohlášení:

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal

  
.....

Bc. Zdeněk Fajkus

Datum odevzdání diplomové práce: 04.05.2012

**Poděkování:**

Chtěl bych poděkovat panu Ing. Davidovi Valovi a panu Ing. Jakubovi Jirkovi za poskytnuté odborné rady a technickou pomoc při řešení diplomové práce. Také bych chtěl poděkovat své rodině za trpělivost a podporu v posledních měsících studia.

## **Abstrakt**

Cílem této diplomové práce je navrhnout řešení pro diagnostické rozhraní elektromobilu a realizovat diagnostický modul. Vzhledem k povaze zařízení se jedná o práci, ve které hraje velkou roli komunikace a množství komunikačních sběrnic. Navrhovaný diagnostický modul disponuje dvěma sběrnicemi CAN, sběrnicí RS232 pro připojení GSM EDGE modemu a Bluetooth modulem. Pomocí kapesního počítače typu MDA je možno komunikovat s diagnostickým rozhraním v elektromobilu a konfigurovat některé periferie jednotky, nebo stahovat soubory s uloženými daty. Tyto soubory lze posléze odeslat na server, který je uloží na diskový prostor počítače. Taktéž lze konfigurovat GSM modem pro připojení k serveru. Na serveru pak běží aplikace pro načítání dat o poloze elektromobilu, kterou si jednotka načítá z GPS modulu připojeného ke sběrnici CAN a pro načítání a ukládání souborů. Diagnostické rozhraní komunikuje se serverem skrze TCP soketovou komunikaci prostřednictvím GSM modemu a MDA skrze Wi-Fi připojení.

## **Klíčová slova**

Texas Instruments, Stellaris, GSM, Bluetooth, TCP komunikace, GPS, MDA, SD karta, CAN, CANOpen

## **Abstract**

The objective of this thesis is to propose solution for the Built-in Electric Car Diagnostic Interface and realization of diagnostic module. Due to character of this facility, where is mean part of communication and communication systems. The proposed diagnostic module has two CAN buses, RS232 bus for connection with GSM EDGE modem and Bluetooth module. With MDA Pocket PC type is possible to communicate with the electric car diagnostic interface and configure some peripheral unit or download files which stored data. These files can be send to server, which saves them to the hard drive of computer. Also is possible to configuration GSM modem for connection with server. On server runs the application to receive data about GPS position of car, which is reading from connected GPS module to CAN bus and for receiving and saving files. Diagnostic interface communicates via TCP socket communication via GSM modem and MDA via Wi-Fi connection.

## **Keywords**

Texas Instruments, Stellaris, GSM, Bluetooth, TCP communication, GPS, MDA, SD card, CAN, CANOpen

## Seznam použitých zkratek

ABS	Anti Block System	NMT	Network Management
ACK	Acknowledge	OD	Object Dictionary
AGM/GEL	Technologie akumulátorů	PC	Personal Computer
ASCII	American Standard Code for Information Interchange	PDA	Personal Digital Assistant
AT	AT příkaz	PDO	Process Data Objects
BMS	Battery Management System	POP3	Post Office Protocol
BUS	Sběrnice	PRO	Professional
CAL	CAN Application Layer	RS232	Sériová linka
CAN	Sběrnice CAN	RS232	Sériová linka
CAN_H	Informační vodič CAN	RTR	Remote Transmission Request
CAN_L	Informační vodič CAN	SD	Secure Digital
CD	Carrier Detect	SD BUS	Secure Digital Bus
CiA	CAN in Automation	SDO	Service Data Objects
CMD	Command	SMD	Surface Mount Device
CMOS	Complementary Metal-Oxide Semiconductor	SMTP	Simple Mail Transfer Protocol
COB ID	Communication Object Identification	SPI	Serial Peripheral Interface
COM	Sériový port	SQL	Structured Query Language
CRC	Cyclic Redundancy Check	SRDO	Safety Relevant Data Objects
CS	Command Specifier	SRR	Substitute remote request
DPS	Deska plošných spojů	SWD	Rozhraní k programování MCU
EDGE	Enhanced Data rates for GSM Evolution	TCP	Transmission Control Protocol
FAT	File Allocation Table	TI	Texas Instruments
FTP	File Transfer Protocol	TTL	Transistor-Transistor Logic
FW	Firmware	UART	Universal asynchronous Receiver Transmitter
GGA	Informace o fixaci pozice	UDP	User Datagram Protocol
GPRS	General Packet Radio Service	USB	Universal Serial Bus
GPS	Global Positioning System	VTG	Vector track an Speed over the ground
GSA	Overall Satellite data	WGS-84	World Geodetic System 1984
GSM	Globale System for Mobile Communication	Wi-Fi	Wireless Fidelity
GSV	Detailed Satellite data		
HTTP	Hypertext Transfer Protocol		
HW	Hardware		
IDE	Identificator Extension bit		
IP	Internet Protocol		
ISO/OSI	International Standards Organization / Open System Interconnection		
JTAG	Join Test Action Group		
LED	Light - Emitting Diode		
Li-Ion	Technologie akumulátoru		
Li-Po	Technologie akumulátoru		
LiFeYPO4	Technologie akumulátoru		
LMT	Local Management		
LSB	Least Significant Bit		
MCU	Micro-cotroller Unit		
MDA	Mobile Digital Assistant		
NGO	Node Guarding Object		
NiCd	Technologie akumulátorů		
NiMh	Technologie akumulátorů		
NMEA	National Marine Electronics Association		

# Obsah

1 Úvod.....	1
2 Problematika elektromobilu.....	2
2.1 Používané akumulátory v elektromobilech.....	2
2.1.1 Akumulátory NiMH.....	2
2.1.2 Akumulátory Li-ion.....	3
2.1.3 Akumulátory LiFeYPO <sub>4</sub> .....	3
3 Popis sběrnice CAN.....	5
3.1 Fyzická vrstva sběrnice CAN.....	5
3.2 Linková vrstva sběrnice CAN.....	8
3.2.1 Datový a ovládací rámec zprávy protokolu CAN 2.0A.....	8
3.2.2 Datový rámec zprávy protokolu CAN 2.0B.....	9
3.2.3 Ovládací rámec protokolu CAN 2.0 A/B.....	10
3.2.4 Chybový rámec.....	10
3.2.5 Zpoždovací (přetěžovací) rámec.....	11
3.2.6 Mezirámcová mezera.....	11
4 Popis komunikačního protokolu CANOpen.....	12
4.1 Pomocné objekty protokolu CANOpen.....	14
5 Návrh řešení diagnostického rozhraní.....	16
5.1 Volba komunikačních modulů.....	16
5.2 Volba paměťového média.....	16
5.3 Uvažovaná funkce zařízení.....	17
5.4 Popis použitého MCU.....	18
5.5 Popis použitého Bluetooth modulu.....	19
5.6 Popis použitého GSM modemu.....	21
5.7 Popis paměťové karty microSD.....	23
5.8 Popis GPS modulu.....	24
6 Realizace diagnostického rozhraní elektromobilu.....	26
6.1 Návrh DPS diagnostického rozhraní elektromobilu.....	27
6.2 Firmware MCU a jeho programování.....	29
6.2.1 Vývoj a popis programu.....	31
6.2.2 Nastavení komunikačních sběrnic.....	32
6.2.3 Komunikace s microSD kartou.....	33
6.2.4 Komunikace s GSM modemem.....	36
6.2.5 Komunikace s Bluetooth modulem.....	39
6.2.6 Komunikace s GPS modulem.....	40
7 Návrh aplikace v mobilním zařízení.....	42
7.1 Komunikace s diagnostickým rozhraním.....	42
7.2 Komunikační protokol diagnostického rozhraní.....	43
7.3 Komunikace mezi MDA a serverem.....	44
7.4 Komunikační protokol serveru.....	46
8 Návrh aplikace na PC – server.....	48
8.1 Komunikace s diagnostickým rozhraním.....	48
8.1.1 Komunikační protokol.....	49
8.1.2 Příjem a dekodování dat.....	49
8.2 Komunikace s mobilní aplikací.....	50
9 Závěr.....	52



# 1 Úvod

V dnešní době konvenčních spalovacích motorů je pojem „elektromobil“ stále častěji objevovaný v různých člancích, zabývající se ekonomikou a ekologií provozu. I výrobci různých mobilních zařízení se angažují při výrobě elektromobilů, především v oblasti komunikací a diagnostiky. Již se objevily elektromobily, které mají integrované komunikační rozhraní se zařízením firmy Apple, na kterém se zobrazují údaje o aktuálních informacích, o rychlosti, dojezdu a podobných údajích, jenž se vyskytují na palubním počítači. Vzhledem k použité technologii je jednodušší využít PDA nebo MDA pro nahrazení palubního počítače, neboť není potřeba vyvíjet nový hardware s patřičným výkonem a nízkou spotřebou, ale pouze vytvořit patřičný komunikační protokol a naprogramovat aplikaci pro dané mobilní zařízení. V dnešním trendu mobilních zařízení je dosahováno již poměrně dostatečného množství výkonu při zajištění malé spotřeby zařízení a dále je i dostatečně rychlá konektivita skrze GSM síť. O podobnou aplikaci se bude jednat i v této práci.

Cílem práce je vytvořit diagnostické rozhraní elektromobilu, komunikující pomocí specifického komunikačního protokolu s ostatními zařízeními jako jsou např. PDA, notebooky či PC a databázový server s SQL databází pro ukládání dat a následném prohlížení v programu na dané platformě.

V první části práce bude seznámení se základní problematikou elektromobilu a porovnání s tradičními automobily. Dále pak s komunikační sběrnici CAN a protokolem CANOpen z důvodu nejčastějšího výskytu této sběrnice v automobilovém průmyslu. Bude popsána jak fyzická vrstva této sběrnice, dosahované rychlosti a možnosti využití pro komunikaci, tak i linková vrstva a její charakteristika pro dané komunikační protokoly. Dále si představíme možnosti bezdrátové komunikace diagnostického rozhraní mezi ostatními zařízeními, která se připojí k jednotce s požadavkem o přenos dat, či on-line sledování a čtení dat.

V druhé části bude hlavní zaměření na hardware a software práce, popis programů a použitých komunikačních protokolů a modulů. V popisu hardwaru se seznámíme s popisem zapojení diagnostického rozhraní a ovládáním celé jednotky. U softwaru si popíšeme ovládání programů a postup jak pracovat s naměřenými daty z elektromobilu.

## 2 Problematika elektromobilu

Kapitola pojednává o základní problematice spjaté s konstrukcí a používáním elektromobilů, jak sériových továrních značek, tak i přestaveb z automobilu se spalovacím motorem prováděných soukromými osobami.

### 2.1 Používané akumulátory v elektromobilech

V elektromobilech se používá sériové spojení článků akumulátorů s kvalitními BMS a různými „inteligentními“ algoritmy dobíjení. Tato kombinace technologií však v minulosti neexistovala, čímž se celkem značně snížil dojezd samotného elektromobilu a hlavně životnost akumulátorů. Při dnešních technologiích je možné dosáhnout přibližně 80 000km s moderní olověnou rekombinační baterií (AGM/GEL) a 150 000km s Li-Po akumulátory a 300 000km s LiFeYPO<sub>4</sub>. [1] Olověné akumulátory se vzhledem k váze, nízké životnosti a malé energetické hustotě v podstatě vytratil z elektromobilové oblasti, avšak ještě stále existují jako trakční baterie v různých vysokozdvížných vozících a podobných pomocných vozidlech. Proto zde představím pouze současně používané akumulátory typu NiMH, Li-ion, LiFeYPO<sub>4</sub>.

[1]

#### 2.1.1 Akumulátory NiMH

S akumulátory typu NiMH se v dnešní době můžeme setkat v hybridním vozidle značky Toyota Prius 1. generace. Katoda je tvořena materiálem hydroxid Niklu a je stejná jako u NiCd akumulátorů. Elektrolytem v akumulátoru je vodný roztok hydroxidu draselného (KOH), který má velmi vysokou vodivost a je zjištěno, že koncentrace elektrolytu zůstává téměř konstantní během nabíjecího či vybíjecího cyklu. Tato vlastnost je velice důležitá, neboť koncentrace elektrolytu je hlavním ovlivňujícím faktorem vnitřního odporu akumulátoru. To znamená, že výkon článku zůstává téměř stejný v celém rozsahu stupně nabití.

Aktivním materiálem záporné elektrody je v podstatě vodík. Ačkoliv není fyzicky možné využít přímo vodík a proto je uložen NiMH článku jako metal hydrid, který slouží jako záporná elektroda. Zajímavostí je, že hydridy kovů zadrží přibližně 1% - 2% vodíku své hmotnosti.

#### Nabíjecí/vybíjecí charakteristiky

Při používání NiMH akumulátorů se projevuje vybíjecí křivka v určité oblasti jako neměnná, což je hlavní rozdíl oproti NiCd akumulátorům, u nichž postupně klesá napětí článku v průběhu vybíjení. Avšak při nabíjení je tento typ akumulátorů velice náchylný na přebíjení, což má za následek snížení kapacity akumulátoru. Proto se využívá několika variant, pro detekci

konce nabíjecího cyklu, nejčastěji se jedná o tzv.  $\Delta U$ , kdy dojde k poklesu napětí článku a tím je indikováno ukončení procesu nabíjení.

#### Samovybíjení akumulátoru

Samovybíjení NiMH akumulátoru je velkým problémem tohoto typu akumulátorů. U některých typů se jedná až o hodnotu 20% kapacity akumulátoru za měsíc. Jedním z důležitých faktorů procesu samovybíjení je okolní teplota, ve které je článek skladován. Je zjištěno, že při vyšší skladovací teplotě je tento proces vyšší a proto se doporučuje umístit tyto typy akumulátorů v chladnějším prostředí po celou dobu uskladnění.

[2]

### **2.1.2 Akumulátory Li-ion**

Li-ion akumulátory jsou v současné době hojně využívány pro napájení mobilních zařízení jako jsou mobilní telefony, fotoaparáty či videokamery. Jejich vysoká hustota energie umožňuje dosažení vyšší kapacity a menší hmotnosti akumulátoru, což v praxi znamená větší časový interval mezi nabíjecími cykly. Katodu tvoří oxid kovu a anoda je tvořena porézním karbonem.

[3]

### **2.1.3 Akumulátory LiFeYPO<sub>4</sub>**

Další verzí Lithium-iontových akumulátorů je Lithium-železo-Yttrium-fosfátové akumulátory (LiFeYPO<sub>4</sub>). Toto označení je dle použitého materiálu na katodě. U anody je použit stejný materiál, jako u ostatních Li akumulátorů, a to uhlík. Jejich hlavní předností je především schopnost dodat vyšší proud a hlavně bezpečnost provozu těchto akumulátorů, jelikož v extrémních podmínkách nevybuchují. Ovšem cenou za tyto vlastnosti je nižší napětí jednoho článku, které činí 3,2V, oproti 3,7V na článek Lithiových akumulátorů.

#### **Výhody:**

- Téměř plochá křivka až do plného vybití akumulátoru.
- Vysoký počet dobíjecích cyklů (2 000 – 3 000).
- Netoxické složení.
- Nemají paměťový efekt.
- Bezpečné oproti jiným Lithiovým akumulátorům.
- Vynikající dobíjecí účinnost (~95%).

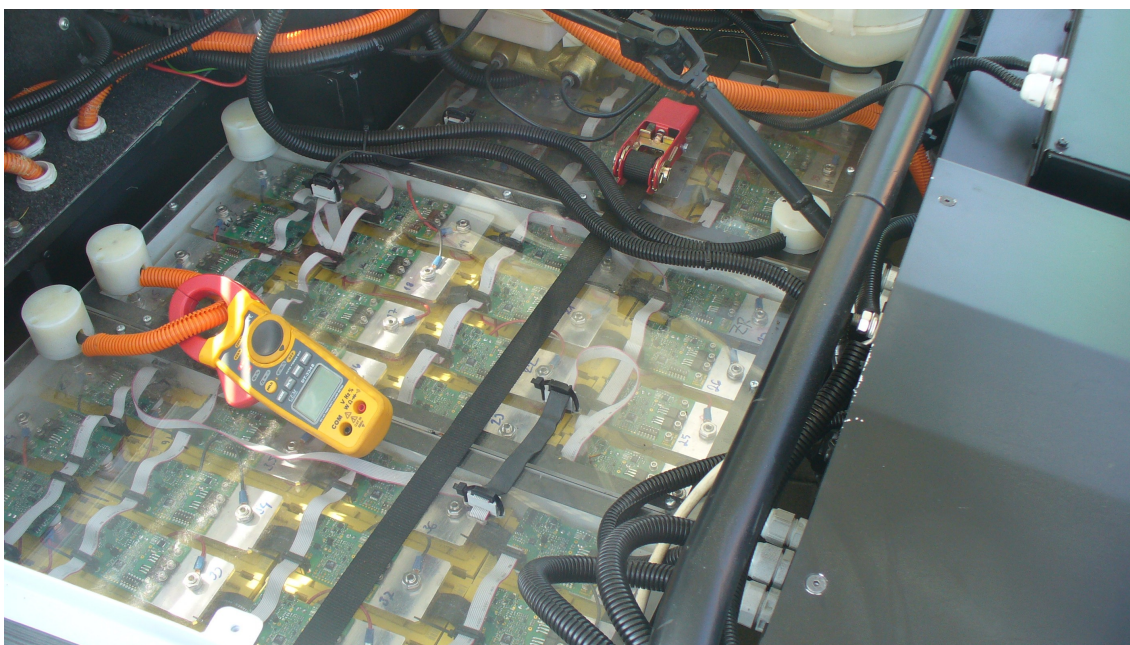
**Nevýhody:**

- Rychlé dobíjení snižuje životnost akumulátoru.
- Možnost předčasného selhání při větším množství hlubokých cyklů (vybití pod 1/3C).

[4]



*Obr. 1: Akumulátorové články LiFeYPO4*



*Obr. 2: Možné uložení akumulátorů LiFeYPO4 v elektromobilu*

### 3 Popis sběrnice CAN

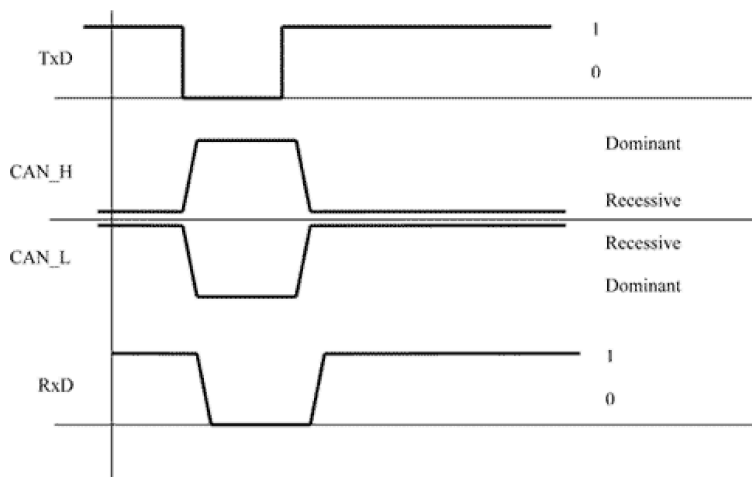
Sběrnice CAN je sériová komunikační sběrnice s definovaným komunikačním protokolem, který podporuje efektivní komunikaci v reálném čase. V automobilovém průmyslu se používá pro připojení různých senzorů k řídicí jednotce, ABS systému a podobných zařízení s rychlostí až 1Mb/s. Pro zajištění jednodušší implementace sběrnice CAN do aplikací byl návrh rozdělen do tří rozdílných vrstev:

- CAN Objektová vrstva.
- CAN Transportní vrstva.
- CAN Fyzický vrstva.

Objektová vrstva a transportní vrstva zahrnuje všechny služby a funkce dat definované linkovou vrstvou ISO/OSI modelu. Rámec objektové vrstvy zahrnuje tyto vlastnosti:

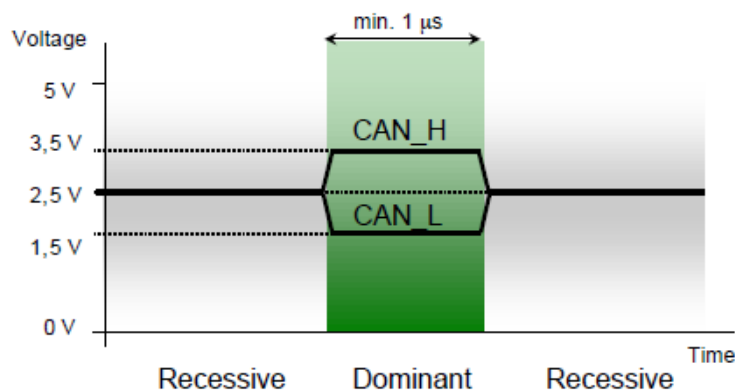
- Zjistit zprávu, která má být odeslána.
- Rozhodnout, která zpráva bude přijata transportní vrstvou.
- Poskytnout rozhraní aplikační vrstvě danému hardwaru.

#### 3.1 Fyzická vrstva sběrnice CAN



Obr. 3: Logické úrovně signálu CAN sběrnice

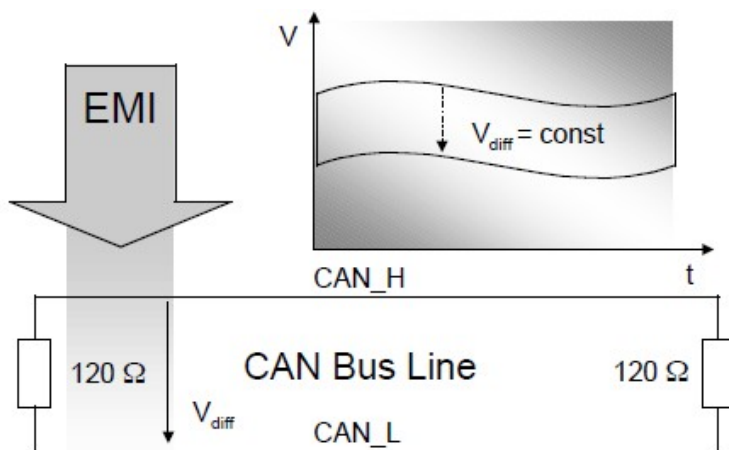
Fyzická vrstva definuje jak jsou jednotlivé bity přenášeny, jejich napěťové úrovně, rychlost přenosu dat a typ přenosového média. Na Obr. 3 jsou zobrazeny bitové logické úrovně signálu CAN sběrnice. Pokud se vysílá log. 1, stav na sběrnici je recesivní, při vysílání log. 0 je stav dominantní. Na Obr. 4 jsou znázorněny napěťové úrovně signálu ve všech třech možných stavech sběrnice.



Obr. 4: Napěťové úrovně signálu

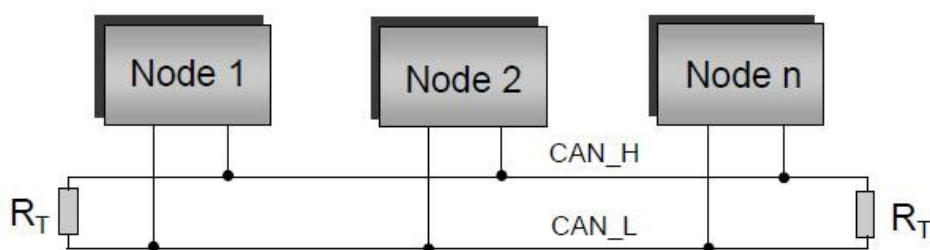
Uzel sběrnice CAN dokáže detekovat recesivní stav, což znamená, že na sběrnici nikdo nevysílá. Recesivní stav je definován velikostí napětí na jednotlivých vodičích CAN\_H a CAN\_L. Napětí na vodiči CAN\_H nesmí překročit napětí  $CAN\_L + 0,5V$ . Jestliže je napětí CAN\_H větší než napětí CAN\_L minimálně o 0,9V, je detekován dominantní stav sběrnice. Nominální hodnoty napěťových úrovní dominantního stavu jsou 3,5V pro CAN\_H a 1,5V pro CAN\_L.

Vzhledem k diferenční povaze přenášeného signálu je sběrnice CAN necitlivá na elektromagnetickou interferenci, jelikož indukce rušivého napětí se projevuje na obou informačních vodičích a na výsledné logické hodnotě signálu se neprojeví, viz. Obr. 5.



Obr. 5: Vliv elektromagnetické interference na informační vedení

### Připojení uzlů na sběrnici



*Obr. 6: Připojení jednotlivých uzlů na sběrnici CAN*

Každý z uzlů připojených na sběrnici se připojí na komunikační vodiče CAN\_H a CAN\_L. Terminační rezistor na koncích sběrnice  $R_T$  má hodnotu  $120\Omega$  a slouží k zamezení odrazů signálu na informačních vodičích.

### Přehled doporučených rychlostí sběrnice

Organizace CAN in Automation (CiA) vydala doporučení o použití rychlostí sběrnice v závislosti na použité délce metalického vedení informačních vodičů. Krom toho je v dokumentu [18] i doporučení na časování bitů, takže uzly jiných výrobců mohou být připojeny na jednu síť CAN bez výpočtu časování bitů, viz Tabulka 1.

Bit rate délka CAN	Nominální délka bitu $t_b$ [ $\mu$ s]	Počet časových kvant na bit	Délka časového kvanta $t_q$	Pozice vzorkovacího bodu
1 Mbit/s 25m	1	8	125ns	$6 t_b$ (750ns)
800 kbit/s 50m	1,25	10	125ns	$8 t_b$ (1 $\mu$ s )
500 kbit/s 100m	2	16	125ns	$14 t_b$ (1,75 $\mu$ s)
250 kbit/s 250m	4	16	250ns	$14 t_b$ (3,5 $\mu$ s)
125 kbit/s 500m	8	16	500ns	$14 t_b$ (7 $\mu$ s)
50 kbit/s 1000m	20	16	1,25 $\mu$ s	$14 t_b$ (17,5 $\mu$ s)
20 kbit/s 2500m	50	16	3,125 $\mu$ s	$14 t_b$ (43,75 $\mu$ s)
10 kbit/s 5000m	100	16	6,25 $\mu$ s	$14 t_b$ (87,5 $\mu$ s )

*Tabulka 1: Doporučení rychlostí a délek sběrnice CAN*

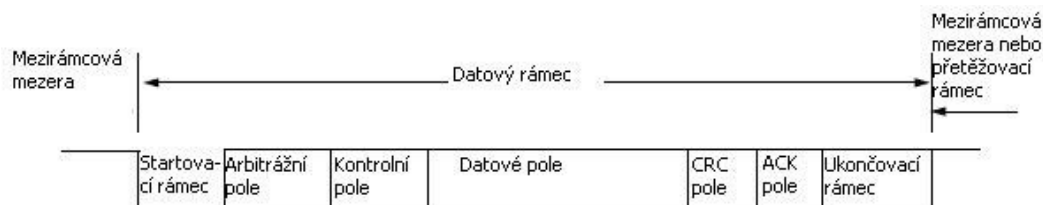
## 3.2 Linková vrstva sběrnice CAN

V komunikačním protokolu CAN 2.0 rozeznáváme čtyři základní rámce:

- Datový rámec – přenáší datové informace po sběrnici.
- Ovládací rámec – přenáší požadavek na přenos dat.
- Chybový rámec – přenáší se při identifikované chybě na sběrnici.
- Zpožďovací (přetěžovací) rámec – generuje zvláštní zpoždění a vkládá jej do předchozích rámců zprávy.
- Mezirámcová mezera – slouží k oddělení jednotlivých rámců.

Komunikace mezi jednotlivými uzly probíhá formou předávání zpráv. Pokud je sběrnice volná, může jakýkoliv uzel vyslat zprávu k jinému adresátu, případně i více adresátům. V případě vysílání zprávy více uzly je komunikace zajišťována podle priority, která je obsažena v adresní oblasti rámce, přičemž nedochází ke ztrátě dat s menší prioritou, pouze se vyčkává na uvolnění sběrnice a následně se data opět odesílají.

### 3.2.1 Datový a ovládací rámec zprávy protokolu CAN 2.0A



Obr. 7: Datový rámec protokolu CAN 2.0

Datový rámec je složen ze sedmi rozdílných bitových polí: Start rámce, Arbitrážní pole, Kontrolní pole, Datové pole, Pole CRC kódu, Pole odpovědi ACK a ukončovacího pole. Datové pole může mít nulovou délku.

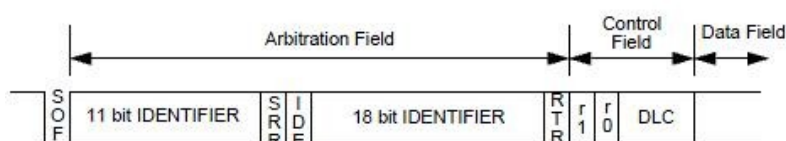
- Startovací pole – označuje začátek datového rámce a Požadavkového rámce. Skládá se z jednoho dominantního bitu.
- Arbitrážní pole – je složeno ze dvou částí – identifikátor a RTR bit. Pro verzi protokolu CAN 2.0A má identifikátor délku 11b, přičemž jsou tyto bity přenášeny v pořadí od ID10 do ID0 s tím, že ID0 je LSB. Bity ID10 až ID4 nesmějí být všechny recesivní. RTR bit udává o jaký typ rámce se jedná. V datovém rámci je tento bit dominantní, kdežto v Požadavkovém rámci je tento bit recesivní.



- Kontrolní pole – skládá se z 6b, přičemž první 2b jsou rezervovány pro další využití a další 4b udávají délku Datového pole.
- Datové pole se skládá z dat, která mají být odeslána v Datovém rámci.
- CRC pole se skládá z CRC kódu a CRC delimiteru. CRC delimiter je v podstatě jeden bit v recesivním stavu.
- ACK pole je složeno ze dvou bitů – ACK SLOT a ACK delimiter. Vysílač odesílá oba tyto bity jako recesivní. Přijímač, který přijme zprávu správně, odešle v ACK SLOT dominantní bit. ACK delimiter je vždy recesivní.
- Ukončovací rámec – označuje konec datového a ovládacího rámce. Tento ukončovací rámec je tvořen sekvencí o délce 7b v recesivním stavu.

### 3.2.2 Datový rámec zprávy protokolu CAN 2.0B

Rozdíl mezi komunikačním protokolem CAN 2.0A a 2.0B je v podstatě pouze v délce Arbitrážního pole obsaženého v rámcích zpráv.



Obr. 8: Popis identifikátoru v protokolu CAN 2.0B

Arbitrážní pole se v protokolu CAN 2.0B skládá z:

- 11b Identifikátor – platí zde stejné pravidlo jako u standardu 2.0A .
- bit SRR (Substitute Remote Request) je odesílán jako recesivní a nahrazuje RTR bit ve verzi CAN 2.0A.
- bit IDE (Identifikator Extension bit) – označuje typ následujícího pole zprávy. Pokud následuje kontrolní pole, je tento bit dominantní a v případě, že následuje pole identifikátoru je tento bit recesivní.
- 18b Identifikátor – je rozšíření předchozího 11b identifikátoru, zvětšuje možnosti adresování.

Další struktura datového rámce a ovládacího rámce je již shodná s protokolem CAN 2.0A.

### 3.2.3 Ovládací rámec protokolu CAN 2.0 A/B

Ovládací rámec je téměř totožný s rámcem datovým, kde je rozdíl pouze v bitu RTR a v tom, že Ovládací rámec nemá datové pole. V případě, že stanice označená jako přijímač požaduje data, vyšle Ovládací rámec stanici, od které tyto data požaduje. Tento Ovládací rámec, jak již jsem psal, neobsahuje datové pole a RTR bit je v recesivním stavu. V kontrolním poli Ovládacího rámce se nachází údaj o délce dat v datovém poli a vzhledem k tomu, že tento rámec nemá datové pole, jsou tyto hodnoty irelevantní.

### 3.2.4 Chybový rámec

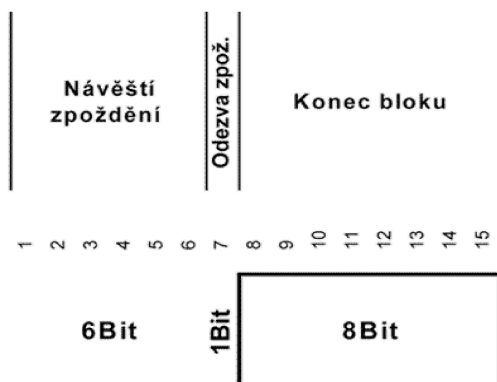
Chybový rámec se skládá ze dvou rozdílných polí. První pole je tzv. Chybový příznak a druhé pole Chybový delimiter. Chybový příznak je dán superpozicí příspěvků z ostatních stanic. Dále se tento příznak rozděluje na dva další typy, a to Aktivní chybový příznak a pasivní chybový příznak.

1. Aktivní chybový příznak se skládá z 6 po sobě jdoucích dominantních bitů.
2. Pasivní chybový příznak se skládá z 6 po sobě jdoucích recesivních bitů, které mohou být přepsány na dominantní jakoukoliv jinou stanicí.

Stanice, která detekuje chybu odesílá Aktivní chybový příznak. Toto způsobí porušení podmínky inverzního 6. bitu k 5. bitu. Ostatní aktivní uzly zjistí porušení podmínky a zahájí odesílání Chybového rámce. Délka tohoto příznaku je v rozmezí 6 – 12 bitů.

Pasivní stanice, která detekuje chybu vyšle pasivní chybový příznak, přičemž ostatní pasivní stanice čekají na sekvenci 6 recesivních bitů. Pasivní chybový příznak je kompletní, pokud je těchto 6 bitů detekováno.

Chybový delimiter se skládá z osmi recesivních bitů odeslaných ihned po sobě.



Obr. 9: Struktura zpoždovacího rámce

### ***3.2.5 Zpožd'ovací (přetěžovací) rámec***

Tento rámec má podobnou strukturu jako chybový rámec. Je využívám k prodloužení odezvy po ukončení datového rámce, ovládacího rámce nebo chybového rámce. V návěští zpoždění se odesílá 6 dominantních bitů. Více bitů není možno odeslat, neboť by to korespondovalo s chybovým rámcem. V odezvě se odesílá jeden dominantní bit a konec bloku obsahuje sekvenci osmi recesivních bitů.

### ***3.2.6 Mezirámcová mezera***

Mezirámcová mezera je složena z bitových polí Zpoždění a nečinnosti sběrnice (BUS IDLE). Pole Zpoždění se skládá ze sekvence tří recesivních bitů a během odesílání tohoto pole je zakázáno kterékoliv stanici odesílat data. Jediná operace, která může být provedena během této události je detekce přetěžovací podmínky. Následující sekvence recesivních bitů je nazvána BUS IDLE a její délka může být až do velikosti Arbitrážního pole datového rámce. Tímto je sběrnice označena jako volná a je umožněno libovolné stanici zahájit vysílání.

[17][16]

## 4 Popis komunikačního protokolu CANOpen

Protokol CAN obsahuje popis pouze prvních dvou komunikačních vrstev ISO/OSI modelu. Protokol CANOpen v podstatě definuje aplikační vrstvu, potřebnou pro dodávání významu datům, přenášeným linkovou vrstvou.

CANOpen rozšiřuje vrstvu CAL definovanou v CiA – 200 a vychází z tzv. „Komunikačního profilu“ definovaného v CiA-301 a CiA-302 a jednotlivé profily zařízení jsou CiA-401 pro digitální a analogové vstupy/výstupy, CiA-402 pro pohony, atd.

Pro protokol CANOpen je vytvořeno několik objektů:

- Adresář objektů (OD – Object Dictionary) – obsahuje obecnou specifikaci zařízení.
- Objekty procesních dat (PDO – Process Data Objects) – jedná se o procesní data, u nichž je čas doručení kritický.
- Objekty servisních dat (SDO – Service Data Objects) – jedná se o servisní data, u nichž čas doručení nemá až tak velký význam. Používají se často u dlouhých zpráv nebo dat pro konfiguraci parametrů zařízení.

Procesní data se přenášejí buď cyklicky, při změně, nebo na vyžádání jako broadcastové zprávy bez dodatečné režie. Rozmístění aplikačních objektů do přenášeného objektu PDO je prováděno pomocí mapování PDO – informace je uložena v adresáři objektů a může být změněna uživatelem. Přenos zpráv SDO je prováděna systémem klient-server, čili se jedná o potvrzovací režim, kdy je každá zpráva potvrzena.

Počáteční nastavení a spuštění celého distribuovaného systému provádí Systém správy sítě – Network Management (NMT). Zprávy CAN obsahují 11-bitový identifikátor, který slouží zároveň jako míra určující prioritu zprávy. Obecně platí, že čím nižší identifikátor je, tím vyšší je priorita zprávy. Tento identifikátor se využívá převážně při kolizi na síti a zpráva s nižší prioritou je zrušena a povoluje se zpráva s prioritou vyšší. Na komunikačním protokolu CANOpen je definován tzv. COB ID (Communication Object ID – viz. Tabulka 2) a příslušný COB ID je přidělován zprávám PDO nebo SDO v adresáři objektů OD (viz. Tabulka 3).

Schéma alokace ID zařízení dovoluje peer-to-peer komunikaci mezi jedním zařízením typu master a až 127 jednotkami typu slave. Také jsou podporovány broadcastové nepotvrzované zprávy NMT služby, synchronizací a objektů časových razítek (Time-Stamp object).

V Tabulka 2 je zobrazena předdefinovaná sada master/slave komunikace, podporující jeden Záchranný objekt, jeden SDO, maximálně 4 přijímací a 4 odesílající PDO a jeden správce sítě NGO (Node Guarding Object). Funkční kód označuje první 4 bity identifikátoru a slouží

jako určování velikosti priority dané přenášené zprávy. Dalších 7 bitů je určeno pro identifikátor, kterým se označuje adresát zprávy. V případě, že je identifikátor roven nule, jedná se o zprávu tzv. Broadcastovou čili určenou všem přijímaným uzlům.

Předdefinovaná sada pro master/slave komunikaci má definovány pouze některé CAN identifikátory, ostatní jsou volné pro další použití. Pokud se připojí jednotka podporující linkování PDO a variabilní alokaci identifikátoru, systém může přiřadit komunikujícímu objektu libovolnou hodnotu identifikátoru. Ovšem s výjimkou identifikátorů určených NMT službě (0), SDO (1405 - 1535 a 1537 – 1663), zprávě kontroly chyb NMT (1793 – 1919).

<b>PtP objekty v předdefinované master/slave komunikaci</b>			
<b>Objekt</b>	<b>Funkční kód (ID10 – ID7)</b>	<b>COB-ID</b>	<b>Komunikační parametr v OD</b>
EMERGENCY	0001	081h - 0FFh	1024h, 1015h
PDO1 TX	0011	181h - 1FFh	1800h
PDO1RX	0100	201h - 27Fh	1400h
PDO2 TX	0101	281h - 2FFh	1801h
PDO2 RX	0110	301h - 37Fh	1401h
PDO3 TX	0111	381f - 3FFh	1802h
PDO3 RX	1000	401h - 47Fh	1402h
PDO4 TX	1001	481h - 4FFh	1803h
PDO4 RX	1010	501h - 57Fh	1403h
SDO TX	1011	581h - 5FFh	1200h
SDO RX	1100	601h - 67Fh	1200h
NMT Error Control	1110	701h - 77Fh	1016h, 1017h
NMT Module Control	0000	0	-
SYNC	0001	80h	1005h, 1006h, 1007h
TIME STAMP	0010	100h	1012h, 1013h

*Tabulka 2: Předdefinovaná sada 4 uzlů peer-to-peer komunikace*

Identifikátory 2015 – 2031 mohou být rezervovány pro NMT, LMT a DBT služby a nemohou být použity k jiným účelům. Také identifikátory 257 – 384 jsou rezervovány pro bezpečnost datových objektů (SRDO – Safety-relevant Data Objects) specifikovaných v rámci CANOpen pro bezpečnou komunikaci.

[18]

Slovník objektů – OD protokolu CANOpen	
Index	Objekt
0000	Nepoužito
0000 - 001F	Statické datové typy (např. BOOL, INT16)
0020 - 003F	Komplexní datové typy (např. PDOCommPar, SDOParameter)
0040 - 005F	Komplexní datový typ specifikující výrobce
0060 - 007F	Specifický statický datový typ profilu zařízení
0080 - 009F	Specifický komplexní datový typ profilu zařízení
00A0 - 0FFF	rezervovaný
1000 - 1FFF	Oblast komunikačního profilu (Název zařízení, počet podporovaných PDO registr chyb, apod.)
2000 - 5FFF	Specifická profilová oblast daná výrobcem
6000 - 9FFF	Standardizovaná profilová oblast
A000 - FFFF	rezervováno

Tabulka 3: Slovník objektů CANOpen

## 4.1 Pomocné objekty protokolu CANOpen

- NMT Control Object - Network Management – Řídicí objekt pro kontrolu a řízení stavu ostatních objektů na sběrnici. Díky nulovému identifikátoru má zpráva nejvyšší prioritu. Zpráva se skládá ze dvou Bytů, z nichž první je *CS* – *Command Specifier* označující druh příkazu a druhý Byte označuje identifikátor uzlu, pro nějž je příkaz určen.

COB-ID	Byte 0	Byte 1
0x000	CS	COB - ID

Tabulka 4: Formát NMT zprávy

Označení příkazu	NMT služba
1	Start ovládaného nódu
2	Zastavení ovládaného nódu
128	Vstoupení do předprovozního režimu
129	Reset zvoleného nódu
130	Reset komunikace

Tabulka 5: Specifikace NMT příkazu

- Node Guarding object – Správce sítě periodicky monitoruje přítomnost a stav Slave zařízení zasíláním Ovládacího rámce. Slave objekt odpovídá zprávou obsahující informaci o jeho stavu a doplňkový bit mění svojí hodnotu při každém dotazu, což umožňuje zjistit, zda došla odpověď je relevantní odpovědí na daný dotaz. Jestliže není odpověď doručena v určitém čase, je tato skutečnost oznámena aplikačnímu programu, který na ní může zareagovat.
- Sync Object – tento objekt slouží pro synchronizaci měření a odesílání hodnot vstupních veličin uzlů.
- Time Stamp Object – Časová značka je objekt poskytující zařízením údaje o aktuálním čase a datu. Zpráva má délku 6 Bytů a obsahuje počet milisekund od půlnoci a počet dní od data 1.1.1984. Na tuto zprávu ostatní uzly neodpovídají.
- Emergency Object – Zpráva je zaslána v případě vzniku závažné chyby v zařízení. Zpráva má vysokou prioritu a její datová část má délku 7 Bytů.

COB-ID	Byte 0-1	Byte 2	Byte 3 - 7
0x80 + Node ID	Chybový kód	Chybový registr (objekt 0x1001)	Výrobcem specifikované chybové pole

*Tabulka 6: Struktura zprávy*

Bit	Typ chyby
0	Generická chyba
1	Proud
2	Napětí
3	Teplota
4	Komunikace
5	Specifický profil zařízení
6	Rezervováno (=0)
7	Specifikováno výrobcem

*Tabulka 7: Obsah chybového registru*

[5] [6] [7]

## 5 Návrh řešení diagnostického rozhraní

Navrhované diagnostické rozhraní elektromobilu by mělo obsahovat několik typů komunikačních standardů a sběrnic, pro připojení různých typů zařízení. Dále je potřebné, aby toto diagnostické rozhraní pracovalo samostatně, bez nutnosti zásahu řidiče, čili musí disponovat dalšími komunikačními moduly a záznamovým zařízením pro off-line diagnostiku provozu vozidla. Při on-line diagnostice, tzn. sledování aktuálních měřených a zaznamenaných dat za provozu, je nutno připojit zobrazovací zařízení, v tomto případě se bude jednat o PDA či MDA s operačním systémem Windows Mobile 6.x PRO. Ovšem připojení ostatních zařízení by neměl být problém, jelikož bude definovaný komunikační protokol s diagnostickým zařízením, čímž by měla být zajištěna funkčnost jiných aplikací s implementovaným komunikačním protokolem.

### 5.1 Volba komunikačních modulů

- MCU s komunikačními sběrnicemi CAN – je požadováno MCU s komunikačními sběrnicemi CAN na úrovni alespoň Linkové vrstvy definované v ISO/OSI modelu.
- Bluetooth modul – Bluetooth modul musí disponovat komunikačním rozhraním SPI nebo UART s rychlostí nejméně 30kB/s. Toto rozhraní bude následně využito k bezdrátovému připojení mobilních zařízení pro off-line nebo on-line diagnostiku dat.
- GPS modul – GPS modul je určen pro zaznamenávání GPS polohy elektromobilu.
- GSM modul – využití tohoto modulu bude převážně jako jeden ze způsobů odesílání dat na server s databází pro ukládání dat z elektromobilu. Dalším možností využití bude kontinuální odesílání dat na výše zmíněný server během jízdy elektromobilu, čili aktuálně změřená a zvolená data budou téměř ihned k dispozici.

### 5.2 Volba paměťového média

Volba paměťového média byla celkem snadná. Vzhledem k dostupnosti, ceně a výkonu byla zvolena paměťová karta formátu microSD. Jednak má dostatečně velkou kapacitu pro záznam dat, tak i rychlost komunikace je plně vyhovující. Také přítomnost souborového systému hrálo velkou roli ve volbě, jelikož je pak celý přístup k datům jednodušší a rychlejší. Jediným problémem je licenční politika Asociace SD, kdy se při výrobě zařízení hostující SD kartu platí licenční poplatek ve výši 1000\$ ročně. Vzhledem k tomu, že se jedná o diplomovou práci, kde bude vyroben jeden kus zařízení, neměl by být s touto licencí problém.



### 5.3 Uvažovaná funkce zařízení

Diagnostické rozhraní elektromobilu by mělo plnit funkci sledování a záznamu dat tekoucích na sběrnici CAN elektromobilu. To znamená, že samotná jednotka připojená ke sběrnici CAN bude v režimu odposlouchávání a zvolená komunikace bude zaznamenávána v určité podobě na SD kartu, pokud si uživatel nezvolí jinak. Dále by měla být dostupná konektivita skrze Bluetooth a GSM modul. Bluetooth bude převážně využit pro komunikaci s mobilním zařízením MDA či PDA nebo notebooku. Tento komunikační kanál bude sloužit k zobrazení dat uložených na SD kartě nebo pro sledování aktuálních dat a jejich okamžitému vyhodnocování. Podobně bude využit i GSM modul, který bude využíván pro komunikaci s databázovým serverem, na kterém bude taktéž možno zobrazovat data uložená na SD kartě, tak i aktuální data načítaná diagnostickou jednotkou. Dále je dostupný GPS modul, který přidává informaci o aktuální poloze elektromobilu k již přijatým informacím po sběrnici CAN. Také je možno využít modulů GPS a GSM jako určitý druh zabezpečení vozidla proti krádeži, kdy je možno zjistit polohu vozidla skrze internet, samozřejmě za předpokladu, že bude GSM modul aktivní, čili v něm bude SIM karta s povolenými daty.

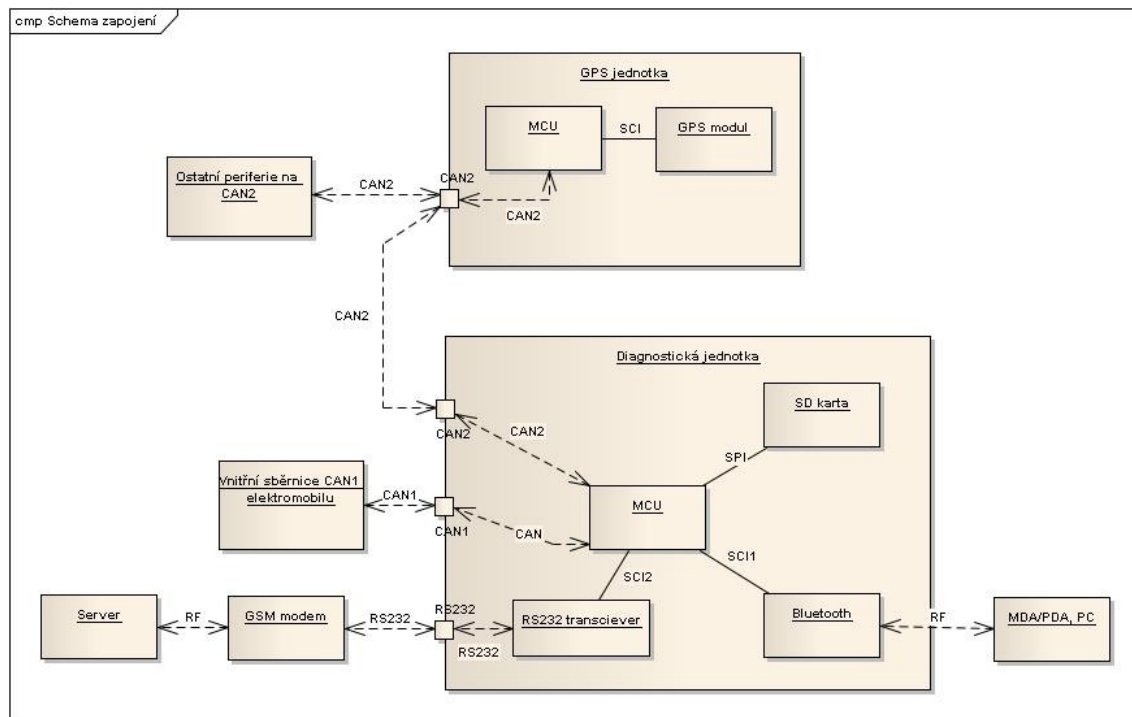
Další částí bude softwarová výbava, která bude obsahovat firmware do zvoleného MCU, aplikaci na MDA s již výše zmíněným operačním systémem a program na PC s operačním systémem Windows XP. Aplikace v MDA by měla umožňovat prohlížet a ukládat zvolená data přijatá po sběrnici CAN, provádět konfiguraci řídicího systému diagnostického rozhraní, případně využívat vlastní GSM a GPS modul místo modulů, implementovaných v zařízení.

Na Obr. 10 je popis zapojení připojení hlavních součástí modulu diagnostického rozhraní elektromobilu. Diagnostické rozhraní je rozděleno na dvě části. První částí je samotná diagnostická jednotka zaznamenávající data elektromobilu ze sběrnice CAN (CAN1) a v druhé části je GPS jednotka spojená s diagnostickým rozhraním prostřednictvím druhé sběrnice CAN. Pro vnější komunikaci na datový server a poskytování údajů z GPS jednotky slouží GSM modem. Rozdělení je nutné z několika důvodů. První je ten, že GPS modul musí mít anténu co nejméně krytou a za druhé, přístup ke GSM modulu musí být co nejlehčí, protože se předpokládá, že SIM karta s datovým tarifem bude občas měněna, čili není možné, aby byl GSM modul schován někde v elektromobilu.

Hlavní diagnostická jednotka obsahuje MCU se sběrnicemi CAN pro zaznamenávání komunikace probíhající v elektromobilu mezi zařízeními a GPS jednotkou. Dále obsahuje SD kartu připojenou skrze SPI komunikační sběrnici a Bluetooth modul komunikující po sběrnici SCI. Bluetooth modul slouží ke komunikaci s mobilními zařízeními skrze virtuální COM port.

GPS jednotka obsahuje MCU od stejného výrobce jako je MCU na diagnostické jednotce, avšak se jedná o typ LM3S2110. V případě připojení druhé sběrnice CAN (CAN2) bude záznam dat probíhat tak, že pro zaznamenávaná data bude vytvořen nový soubor

a pro GPS pozici bude k dispozici taktéž nový soubor.



Obr. 10: Blokové schéma zapojení

## 5.4 Popis použitého MCU

Při výběru vhodného MCU bylo přihlášeno na několik kritérií výběru, které rozhodly o využití MCU firmy Texas Instruments rodiny Stellaris s jádrem ARM Cortex M3. Jedná se o typ LM3S2965. Kritéria byla následující:

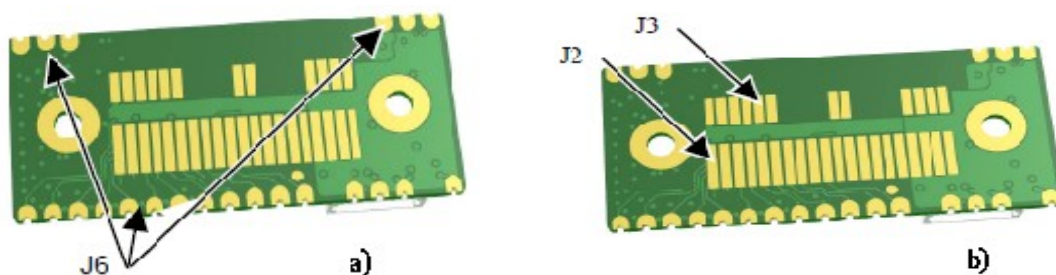
- Cena.
- Výkon MCU v závislosti na spotřebě jádra.
- Dostupnost komunikačních sběrnic: 3x SCI, 1x SPI, 2x CAN 2.0 A/B.
- 32-bit jádro MCU.
- Napájení 2,7V – 5,5V.
- Větší paměť RAM .

Tento typ vyhovoval ve všech kritériích a to následujícími výsledky:

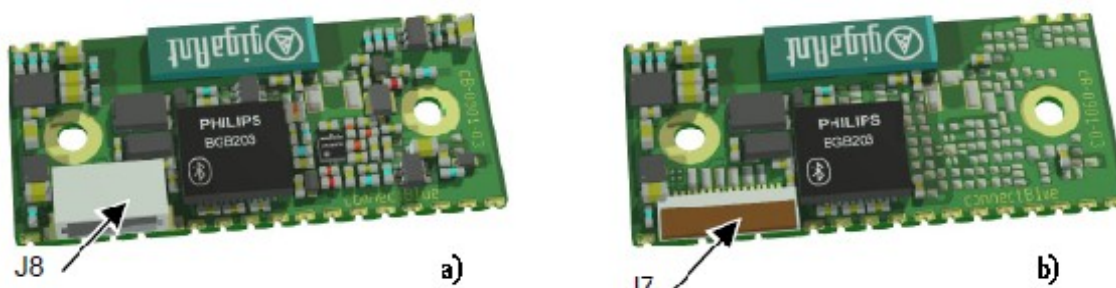
- Cena: přibližně \$4,65 při odběru 1000ks.
- Výkon MCU: taktovací frekvence může být až 50MHz.
- Dostupnost sběrnic: 3x SCI, 1x SPI, 2x CAN 2.0A/B.
- 32-bit jádro MCU: Splněno.
- Napájení 2,7 – 5,5V: Splněno.
- Velikost RAM: 64kB.

## 5.5 Popis použitého Bluetooth modulu

Použitý Bluetooth modul je od společnosti connectBlue AB s maximálním vysílacím výkonem 49mW a interní vestavěnou anténou. Bluetooth modul má několik variant konektorů pro připojení k zařízení viz. Obr. 11 a), b) a Obr. 12 a), b).



Obr. 11 a), b): Pájecí konektory



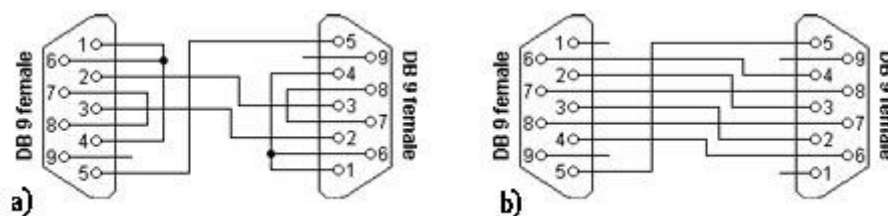
Obr. 12 a), b): Konektory na JST a flexfilm kabel

Pro tuto aplikaci jsem vybral možnost připojení pomocí konektoru J6 na Obr. 11 a). Jedná se o pájecí typ a touto možností připojení je i samotný modul fixován na patřičné pozici

na DPS.

Napájení Bluetooth modulu je dovoleno v rozsahu 3,0 až 6,0V, přičemž typická hodnota napájecího napětí je 3,3V a hodnoty logických úrovní jsou tedy v CMOS logice. Dále tento modul umožňuje řízení spotřeby pomocí AT příkazů, které jsou taktéž využity pro kompletní řízení Bluetooth modulu. K tomuto modulu lze připojit externí anténu s montáží na konektor nebo koaxiální kabel.

Pro konfiguraci modulu můžeme využít program Hyperterminal společnosti Microsoft, avšak je nutné mít převodník logických úrovní signálu RS232 – UART. Při prvním použití Bluetooth modulu je nutné mít nastavené HW řízení toku dat, nebo propojit tyto piny dle standartu RS232. Na Obr. 34 je znázorněno propojení pinů sběrnice UART nastavené pro trojvodičové propojení sběrnice, přičemž je minimálně jedno zařízení nastaveno na HW řízení toku dat.



Obr. 13: a) Obvyklé zapojení RS232  
b) Zapojení RS232 - plné řízení toku

Tabulka 8 Obsahuje popis jednotlivých pinů, jejich název a číslo pinu na konektoru Canon 9. Pro komunikaci mezi zařízeními se nejčastěji využívá křížených kabelů, které jsou zapojeny stejně jako na Obr. 35.

[14]

Číslo, popis a vlastnost jednotlivých pinů		
1+6	Data Set Ready + Carrier Detect	Připraven přijímat data a detekce nosné
2	Receive Data	Příjem dat
3	Transmit Data	Vysílání dat
4	Data Terminal Ready	Terminál je připraven přijímat
5	System Ground	Systémová zem
7	Request to Send	Požadavek pro vysílání dat
8	Clear to Send	Připraven vysílat data

Tabulka 8: Zapojení konektoru a popis pinů RS232

## 5.6 Popis použitého GSM modemu

Použitý GSM modem je vybrán od firmy Sectron a jedná se o typ MC75i EDGE Terminal. Modul je zapouzdřen v hliníkovém šasi a je vybaven externí anténou se ziskem 2dB. S tímto modemem se komunikuje výhradně pomocí AT příkazů, které jsou rozděleny do skupin dle svého využití. Pro využití daného modemu jsou klíčové tyto vlastnosti:

- Čtyřpásmová transciever na GSM frekvence: 850, 900, 1800 a 1900MHz.
- Komunikace ve standardu EDGE12 a GPRS12.
- Statusová LED.
- Široký rozsah napájecího napětí – od 8VDC až 30VDC s příkonem 5W.
- Výkonový rozsah: - Class 4 (2W) GSM850/900 a Class 1 (1W) GSM1800/1900.
- Internetové služby: TCP, UDP, HTTP, FTP, SMTP, POP3.
- Hodnota krytí: IP42.

V připojení na síť v režimu EDGE dosahuje modem maximální přenosové rychlosti 236,8 kb/s v obou směrech (upload/download). V režimu GPRS je tato rychlost maximálně na 86kb/s v obou směrech toku dat.

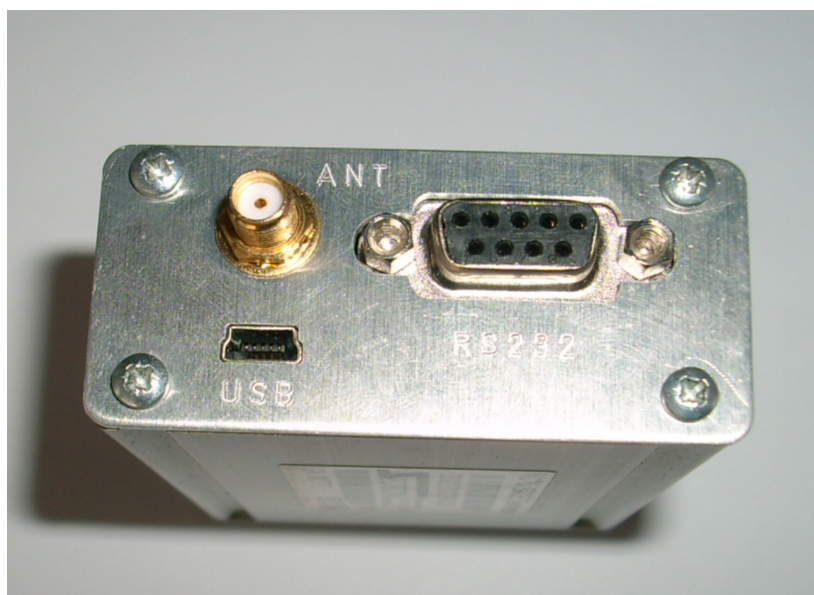
Komunikační rozhraní jsou na GSM modemu k dispozici celkem dvě: USB a sériový port. V této aplikaci se bude využívat pouze sériový port s továrním nastavením komunikačních parametrů:

- Komunikační rychlost 115 200 b/s.
- 8 datových bitů, 1 stop bit.
- Ochrana paritou: žádná.
- Bez řízení toku dat (HW, SW).

[22]

Na Obr. 14 je vyobrazen GSM modem ze strany připojení komunikačních rozhraní. Na pravé straně je konektor typu Canon 9 pro připojení sériové linky, v levém horním rohu je SMA konektor pro připojení externí antény a pod tímto konektorem se nachází USB 2.0 port pro připojení k PC.

Na Obr. 15 je znázorněna část příslušenství GSM modemu. Při postupu z levé strany se jedná o: externí anténu, GSM modem a RS232 kabel pro připojení k PC nebo vlastní aplikaci.



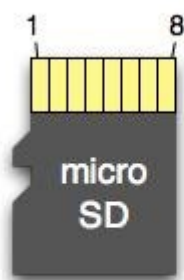
*Obr. 14: GSM modem - komunikační rozhraní*



*Obr. 15: GSM modem včetně příslušenství*

## 5.7 Popis paměťové karty microSD

Zvolené paměťové médium disponuje dvěma typy komunikačních rozhraní. Jedná se o standardní komunikační sběrnici SPI a sběrnici SD Bus. Na Obr. 16a) je znázorněno číselné označení pinů SD karty, které je stejné u všech dostupných formátů SD karet.



a)



b)

Obr. 16: a) Číslování pinů microSD karty

b) microSD karta

[23]

Pin	Název pinu	Popis signálu SDBUS mód	Popis signálu SPI mód
1	DAT3/CS	Datový vodič 3	Výběr zařízení - Slave Select (SS)
2	CMD/DI	Indikace příkazu CMD	Master Out/Slave In (MOSI)
3	GND	Uzemnění	Uzemnění
4	VCC	Napájení (typ. 3,3V)	Napájení (typ. 3,3V)
5	CLOCK	Hodinový signál	Hodinový signál (SCK)
6	GND	Uzemnění	Uzemnění
7	DAT0/D0	Datový vodič 0	Master In/Slave Out (MISO)
8	DAT1/IRQ	Datový vodič 1	Nepoužitý nebo pro IRQ
9	DAT2/NC	Datový vodič 2	Nepoužitý

Tabulka 9: Popis pinů SD karty v obou režimech komunikace

V Tabulka 9 je popis pinů microSD karty pro oba dostupné komunikační režimy a na Obr. 16a) je číslování pinů. Pro tuto aplikaci je vhodné použít režim komunikace po sběrnici SPI, jelikož touto sběrnici disponuje použitý mikroprocesor a není potřeba využívat plné rychlosti komunikace s SD kartou, kterou disponuje režim SDBUS. Při návrhu aplikace

s SD kartou je nutné dodržet rozsah napájecího napětí v rozmezí 2,7 až 3,6V, přičemž maximální zvlnění napájecího napětí činí 60mV. Spotřeba proudu SD karty je rozdělena do tří kategorií, kde maximální hodnoty jsou:

- Režim spánku: 250  $\mu$ A.
- Režim čtení dat: 65 mA.
- Režim zápisu dat: 75 mA.

[13][9]

## 5.8 Popis GPS modulu

Použitý GPS modul byl vybrán od firmy Leadtek, model LR9552 TTL. Označení TTL u tohoto modulu znamená, že logické úrovně sběrnice UART jsou v TTL logice.

Parametr	Popis
Rychlost UARTu	4800 (default), 9 600, 19 200, 38 400, 57 600
Protokol	NMEA-0183 ASCII
Formát data	WGS-84
Formáty zpráv	GGA(1s), GSA(5s), GSV(5s), RMC(1s), VTG(1s)
Výstupní frekvence	1Hz

*Tabulka 10: Přehled hlavních parametrů GPS modulu*

Veličina	Hodnota
Pozice	10m, 5m s korekcí WAAS
Rychlost	0,1 $\text{ms}^{-1}$
Čas	1 $\mu$ s – synchronizace s GPS časem

*Tabulka 11: Přesnost GPS modulu*

V Tabulka 10 je přehled základních parametrů GPS modulu udávaných výrobcem v manuálu. V následující Tabulka 11 jsou uvedeny přesnosti pro zmíněné veličiny.

[20]

Korekce WAAS je dostupná pouze ve Spojených státech amerických a jedná se o síť přibližně 25 pozemních stanic, které monitorují satelitní GPS data. Dvě hlavní stanice, nacházející se každá na jednom z pobřeží, shromažďují data z referenčních stanic a vytvářejí



korekční GPS zprávu z časového posunu a zpoždění signálu vlivem atmosféry a ionosféry. Tuto zprávu šíří prostřednictvím geostacionárního satelitu nebo satelitu se statickou polohou nad rovníkem.

[21]

## 6 Realizace diagnostického rozhraní elektromobilu

V této kapitole je představena demonstrace realizace samotného diagnostického rozhraní spolu se schématy a diagramy aktivit, případně vývojovými diagramy. Je zde popsána problematika propojení komunikačních sběrnic a modulů, které jsou využity v diagnostickém rozhraní.

Centrálním mozkiem diagnostického rozhraní je již výše zmíněný MCU firmy TI, který je fyzicky propojen s Bluetooth modulem, SD kartou, CAN transciervy a RS232 převodníky. Připojený Bluetooth modul je nastaven z továrního nastavení a pro ulehčení manuální práce při výrobě DPS jsou propojeny vodiče pro hardwarové řízení toku dat sběrnice UART. Tato funkce lze sice vypnout skrze AT příkazy, ale tovární nastavení má nastavené HW řízení toku dat. Dále je na SPI sběrnici připojena microSD karta sloužící pro záznam komunikace na obou sběrnicih CAN mezi zařízeními. Na jeden UART port MCU je také připojen RS232 převodník firmy MAXIM, který má napájení 3,3 V a převádí logické úrovně sběrnice RS232 a UART. Na tento UART je připojen GSM modem, který vytváří spojení do internetu. Celá tato jednotka je napájena spínaným zdrojem LM2672 s napětím 3,3V a s maximálním proudovým zatížením o velikosti 1A. Spínaný zdroj je zvolen z důvodu efektivity a účinnosti zdroje oproti zdrojům lineárním.

Při spuštění jednotky dochází k detekci dostupnosti microSD karty a následné inicializaci, jak samotné paměťové karty, tak vlastního souborového systému FAT32 obsaženého na kartě. Dále se provádí detekce dostupnosti GSM modemu. Jakmile je GSM modem detekován, testuje se možnost registrace na GSM síť a dostupnost datového připojení. Testování Bluetooth modulu je posledním krokem inicializace jednotky. Zde se testuje pouze fyzická dostupnost modulu a schopnost odpovědět na AT příkaz. Inicializace paměťového média probíhá jako první, neboť během inicializace ostatních periférií je již možno ukládat data do vnitřního zásobníku MCU a následně zapsat data na microSD kartu do souboru. Jednotka disponuje dvěma sběrnicih CAN, kde se jedna může využít jako systémová sběrnice, na které probíhá komunikace mezi systémovou částí elektromobilu, jako jsou frekvenční měniče nebo řídicí systém akumulátoru. Druhá sběrnice CAN může být napojena na ostatní zařízení, které neslouží k zabezpečení provozu elektromobilu. V tomto případě je na druhé sběrnici CAN připojen GPS modul a dochází k periodickému přijímání dat o poloze vozidla a to s periodou 1s. GPS modul je vytvořen jako demonstrační aplikace a GPS data přijatá na této sběrnici jsou ukládána do souboru s názvem *gps.txt*.

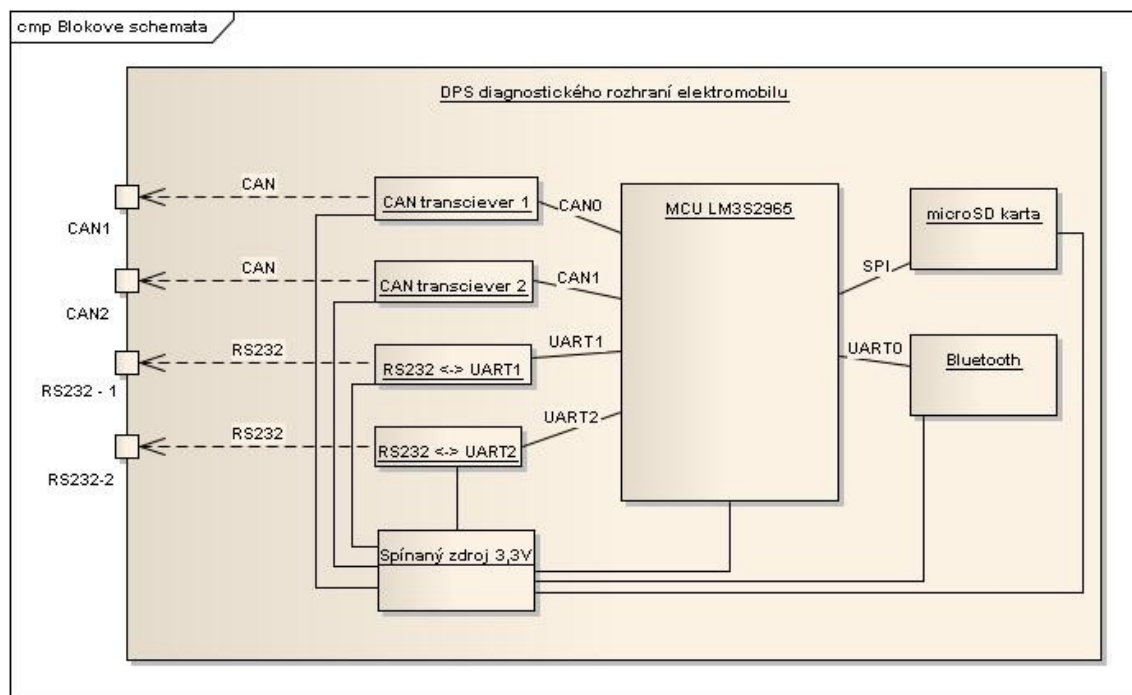
Před spuštěním aplikace v MDA je nutné, aby bylo zařízení spárováno s MDA. Bez tohoto spárování nebude možné se s jednotkou spojit, protože komunikace probíhá skrze virtuální COM port, který není vytvořen do doby, než se obě zařízení spárují. Samotná aplikace v MDA je založena na komunikaci a je spíše určená pro dodatečnou konfiguraci diagnostického rozhraní v elektromobilu. Je možné načítat dostupná data z elektromobilu, stejně jako stahovat

soubory z elektromobilu. Také je vytvořena SQL databáze, do které je možno zapisovat aktuální data.

Aplikace na PC je vytvořena jako serverová aplikace pro on-line sběr dat z elektromobilu. Přijatá data rovněž ukládá do databázového souboru a je možno si tyto data zobrazit v textové podobě. Skrze TCP komunikaci sbírá data z GSM modemu a je nutné, aby měl počítač přidělenou veřejnou IP adresu a ověřit si dostupnost portů, které budou používány při komunikaci.

## 6.1 Návrh DPS diagnostického rozhraní elektromobilu

Při návrhu zapojení desky plošných spojů diagnostického rozhraní bylo vycházeno z blokového schématu zapojení z Obr. 17. Hlavním obvodem je zde MCU LM3S2965, ke kterému jsou připojeny obvody pro komunikaci (CAN transcievery, převodníky logických úrovní sběrnice UART na RS232 a zpět a Bluetooth modul) a paměťová karta formátu microSD. Celá jednotka je napájena spínaným zdrojem 3,3V.

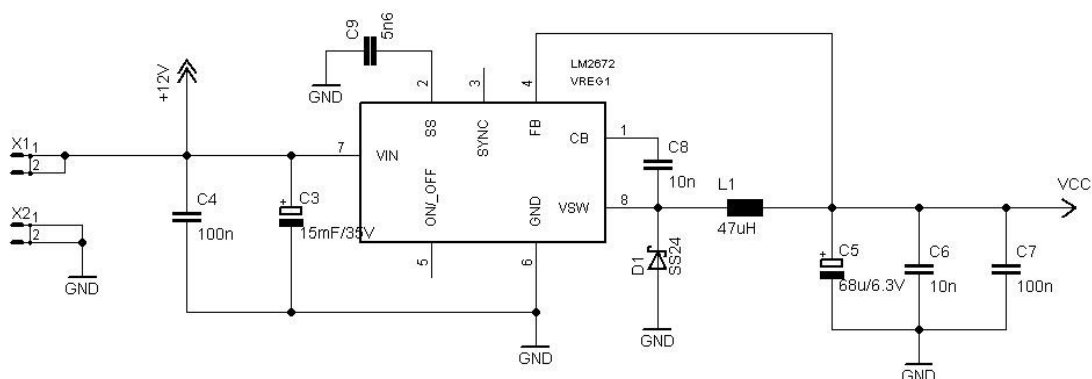


Obr. 17: Blokové schéma zapojení diagnostického rozhraní

Při návrhu DPS diagnostického rozhraní elektromobilu se využilo doporučených zapojení dle doporučení výrobců jednotlivých součástí, obvodů a modulů. Je využit spínaný zdroj firmy National Semiconductor s typovým označením LM2672-3.3. Jedná se o integrovaný obvod spínaného zdroje včetně DMOS tranzistoru. Dosahuje účinnosti až 96% a bylo vybráno

pouzdro s označením SO8.

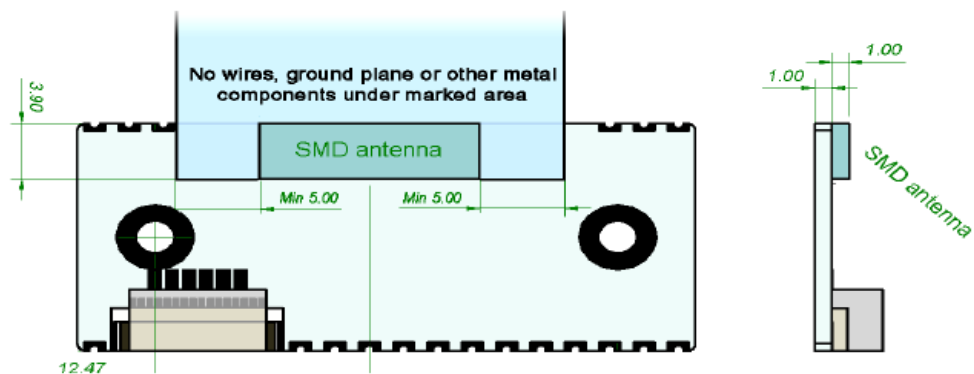
Na Obr. 18 je schéma zapojení spínaného zdroje LM2672-3.3 s výběrem součástek vycházejících z dokumentace výrobce [10] a jeho doporučeného návrhu parametrů součástek. Výstupní napětí z tohoto zdroje je po zapojení naměřeno na hodnotě 3,29V, přičemž maximální přípustná změna napětí je při maximální změně zatížení 100mV. Z toho vyplývá, že i při velké změně odběru je toto napětí v rozsahu napájecích napětí všech součástí obvodu a nemělo by dojít k destrukci zařízení.



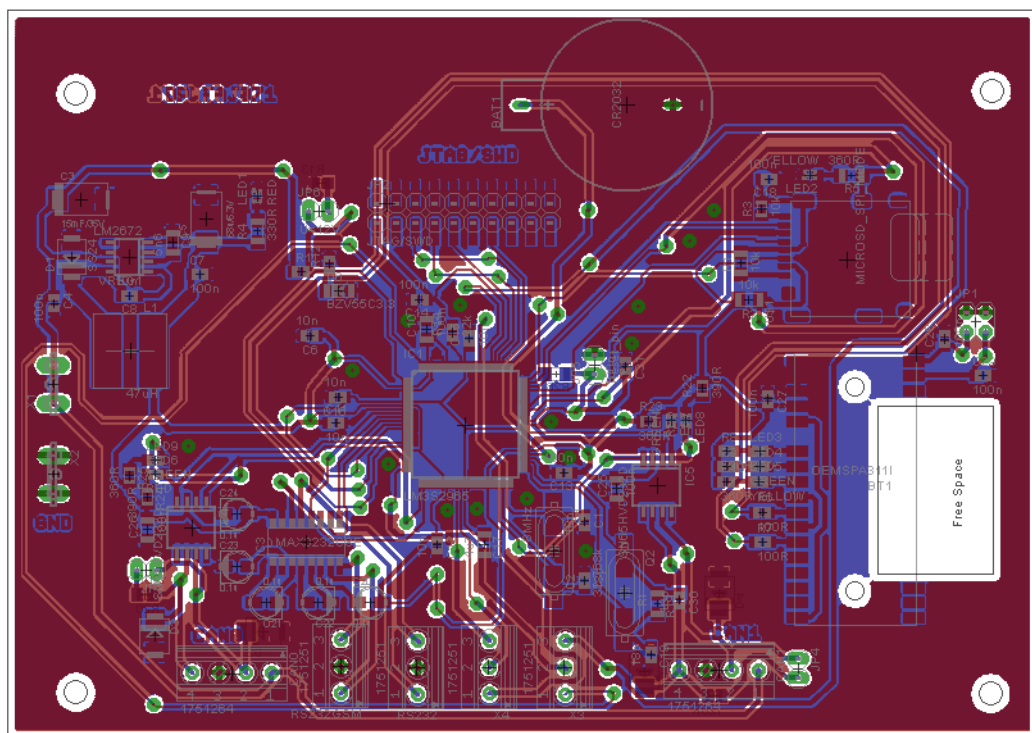
Obr. 18: Schéma zapojení spínaného zdroje LM2672-3.3

Dále bylo nutné dodržet vymezení prostoru u antény Bluetooth modulu (viz. Obr. 19), kterou doporučuje výrobce v dokumentaci k modulu [12]. Dle tohoto doporučení se v prostoru antény nesmí nacházet žádná vodivá plocha, ani informační či napájecí vodiče. Dále se nesmí vyskytovat libovolná kovová plocha v minimálním okruhu 10mm od integrované SMD antény. V opačném případě by docházelo k výraznému zhoršení funkce Bluetooth modulu, případně až k nemožnosti bezdrátově komunikovat s modulem.

Kompletní schéma zapojení se kvůli své velikosti nachází v Příloha 1. Návrh DPS v programu Eagle je znázorněn na Obr. 20. Je zde vidět volné místo u SMD antény Bluetooth modulu a vynechání místa u cívky spínaného zdroje z důvodu rušení ostatních součástí obvodu. Zapojení ostatních prvků je převzato z dokumentací a doplněno o filtrační kondenzátory s hodnotou 100nF pro odfiltrování vysokofrekvenčních složek napájecího napětí.



Obr. 19: Popis vymezení prostoru v oblasti antény Bluetooth modulu



Obr. 20: Návrh DPS v programu Eagle

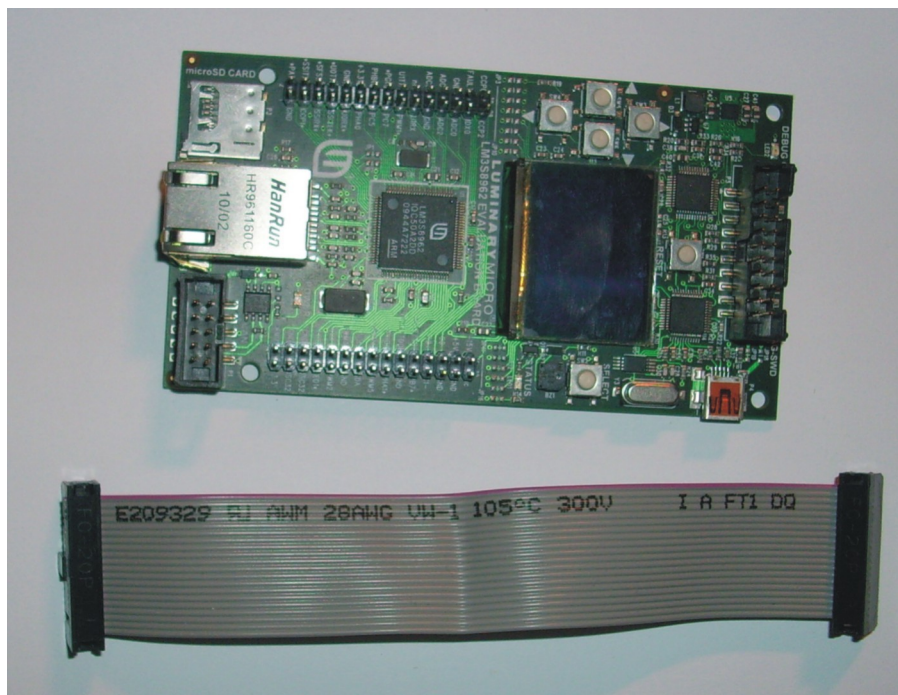
## 6.2 Firmware MCU a jeho programování

Při tvorbě FW pro použitý MCU bylo využito programovací prostředí firmy Texas Instruments s názvem Code Composer Studio verze 5.1.1.00028 s plnou licencí. Licence je ale

pouze testovací, a to na 90 dní, poté se produktová licence zruší a aktualizuje se na standardní typ pro nekomerční využití s limitací kódu. Pro nahrání programu do MCU byl využit Development Board s MCU Stellaris LM3S8962 a dostupným programovacím rozhraním SWD/JTAG, které slouží pro nahrání uživatelského programu do externí DPS. Na Obr. 21 je vyobrazen vývojový kit od firmy TI a programovací, 20-ti žilový plochý kabel pro připojení k cílové aplikaci. Taktéž je možno využít program LM Flash Programmer, který naprogramuje MCU za použití tohoto vývojového kitu.

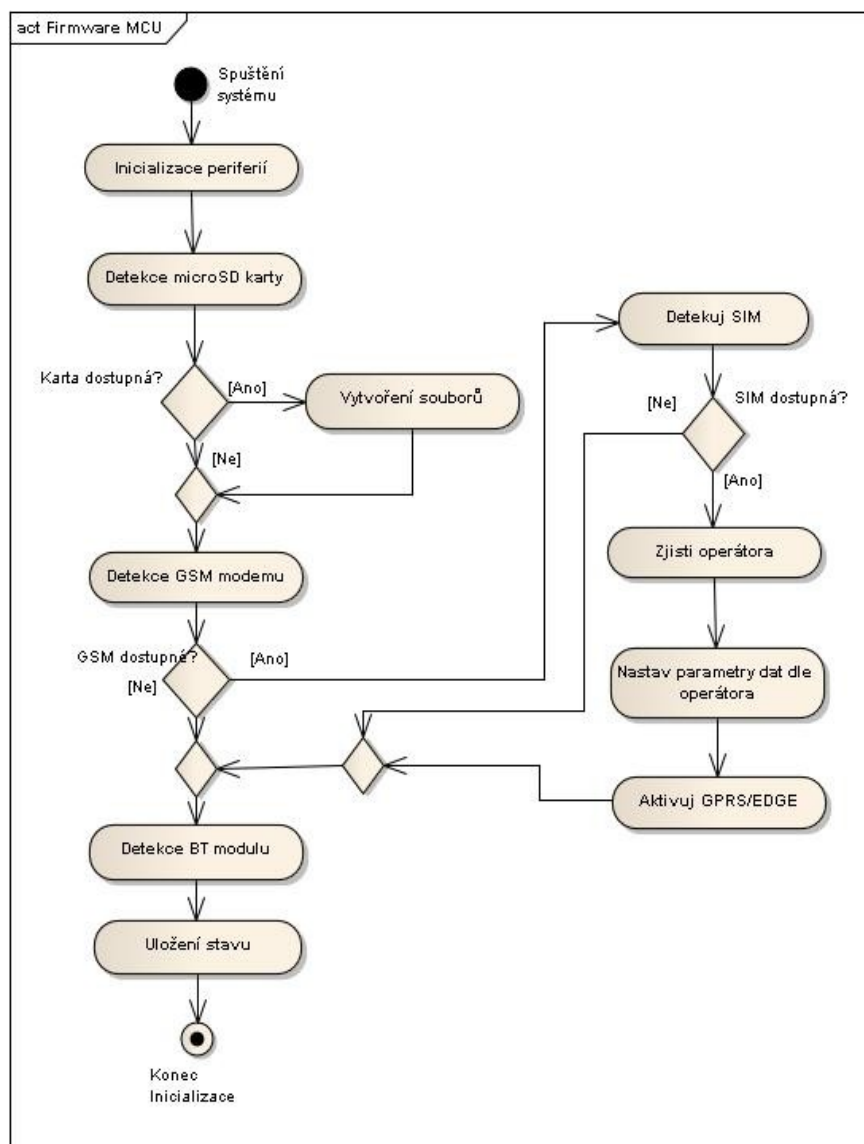
Při vytváření aplikace pro MCU jsou využívány knihovny výrobce, tzv. *StellarisWare*. V těchto knihovnách jsou obsaženy obslužné rutiny pro většinu periférií obsažených v MCU, také je možno využít příkladů jak pro periférie, tak pro celé projekty. Některé funkce budou popsány dále. Pro možnost využití těchto knihoven jsou dvě varianty. Většinou jsou jednotlivé knihovny vázány na ostatní a je nutné v jednotlivých souborech přepsat cesty ke každému souboru zvlášť, nebo je v možno vložit cestu ke knihovnám do nastavení projektu v sekci *Include Option*. Pakliže zvolíme druhou a jednodušší variantu, přidáme cestu *C:\StellarisWare* do oblastí, které se mají prohledávat při kompilování projektu.

Před nahráním vlastního programu do MCU je nutné do svého projektu přidat soubor s názvem *startup\_css.c*, který obsahuje kód obsluhující přerušovací rutiny a restart jádra MCU po zapnutí. Tento soubor je dostupný v kterémkoliv vzorovém projektu výrobce a také na CD jako příloha k této diplomové práci.



Obr. 21: Programátor pro MCU Texas Instruments

### 6.2.1 Vývoj a popis programu



Obr. 22: Vývojový diagram inicializace MCU a periférií

Vývoj programu probíhal dle navrženého schématu činnosti z Obr. 22. Schéma znázorňuje posloupnost kroků při inicializaci. Nejprve je nutné inicializovat periférie, tzn. komunikační sběrnice a vše co je nutné k jejich správné funkci. Po úspěšné inicializaci periférií začíná proces detekcí a inicializací modulů mimo MCU. Prvním procesem detekce a inicializace je načtení microSD karty, hlavně díky možnosti, že se již v době inicializace ostatních periférií může na libovolné sběrnici CAN objevit datový proud. Primárně se vše nejprve ukládá do bufferu, ale při výskytu většího množství dat již může být buffer plný a došlo by ke ztrátě dat. V případě, že je microSD karta detekována, vyhledají se dostupné soubory a

následně se vytvoří soubory pro záznam dat, do kterých se bude později zapisovat. Obdobnou inicializací prochází všechny moduly a následné parametry se uloží do paměti MCU pro její následné čtení, buďto na požadavek obsluhy MDA nebo samotných procedur MCU.

### 6.2.2 Nastavení komunikačních sběrnic

Nastavení všech potřebných periférií MCU bylo vytvořeno dle příručky [19] výrobce MCU. Nejprve bylo nutné nastavit zdroj hodinového signálu pro MCU, jinak by periférie nepracovaly správně. Toto nastavení bylo provedeno funkcí *SysCtlClockSet(params)*, viz. níže.

```
SysCtlClockSet(
    SYSCTL_SYSDIV_1 | SYSCTL_USE_OSC | SYSCTL_OSC_MAIN
    | SYSCTL_XTAL_8MHZ);
```

Parametr	Význam
SYSCTL_SYSDIV_1	Nastavení děličky vstupního kmitočtu
SYSCTL_USE_OSC	Informování MCU o použití externího zdroje
SYSCTL_OSC_MAIN	Informování MCU o prioritním využití externího zdroje
SYSCTL_XTAL_8MHZ	Hodnota krystalu

Tabulka 12: Parametry nastavení zdroje hodinového signálu

Vzhledem k využití sady knihoven od výrobce je nastavení periférií mnohem jednodušší, než je tomu u jiných výrobců, kde je nutné hledat jednotlivé registry a jejich význam. Pro příklad zde uvedu nekomentovaný kus kódu pro nastavení sběrnice CAN. Nekomentovaný proto, aby bylo vidět, jak je tato operace jednoduchá.

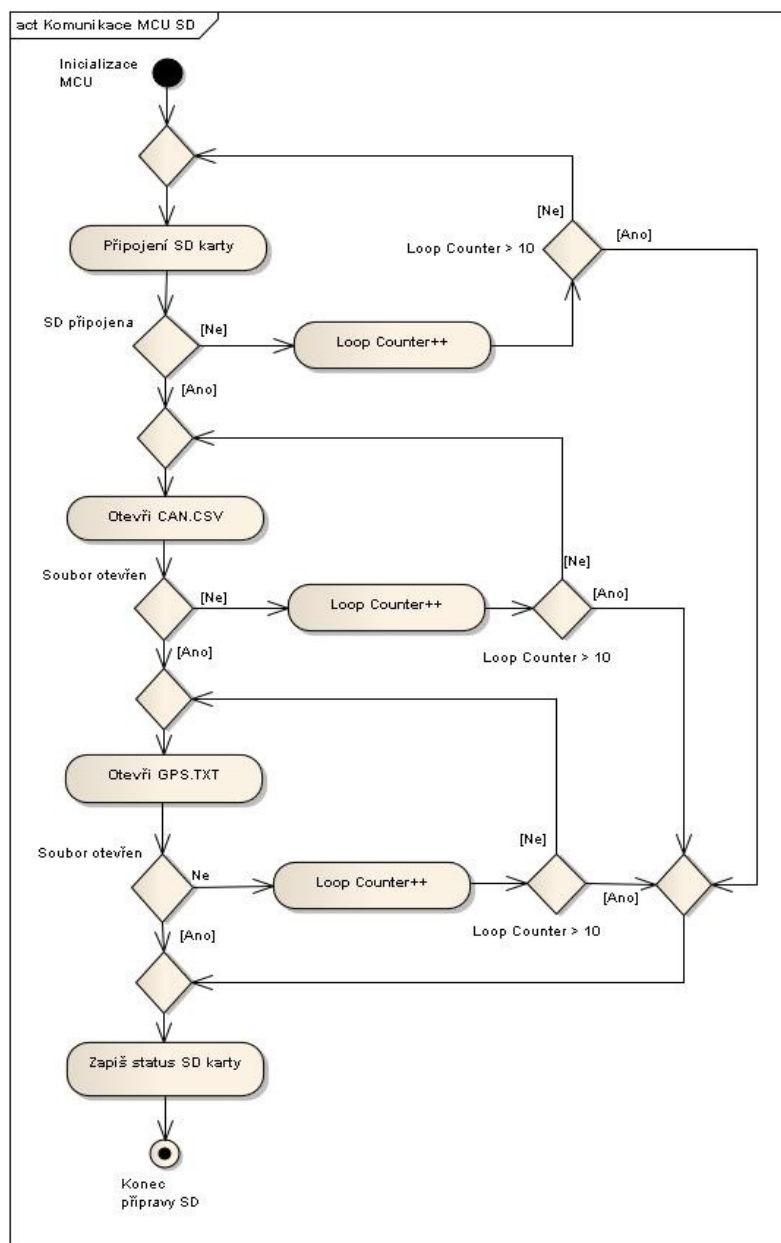
```
SysCtlPeripheralEnable(SYSCTL_PERIPH_CAN1);
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
GPIOPinTypeCAN(GPIO_PORTF_BASE, GPIO_PIN_0 | GPIO_PIN_1);
CANInit(CAN1_BASE);
CANBitRateSet(CAN1_BASE, 8000000, 250000);
CANEnable(CAN1_BASE);
CANIntRegister(CAN1_BASE, CAN2dataRX);
CANIntEnable(CAN1_BASE, CAN_INT_MASTER | CAN_INT_STATUS);
IntEnable(INT_CAN1);
```

Nastavení ostatních sběrnic je velice podobné, resp. funkce využitě pro nastavení se příliš neliší od těchto vzorových. V příručce k těmto knihovnám jsou i příklady nastavení a využití těchto funkcí.



### 6.2.3 Komunikace s microSD kartou

Pro komunikaci s microSD kartou je využita sběrnice SPI a dostupné knihovny pro práci se souborovým systémem FAT32 obsaženým na paměťové kartě. Všechny potřebné operace pro komunikaci jsou obsaženy v knihovnách „ff.c“ a „diskio.h“. Pro správnou funkci ovladačů FAT systému je nutné definovat piny, které jsou využité při komunikaci. Tyto definice jsou uloženy v přílohách na CD v souboru *fat.c*.



Obr. 23: Vývojový diagram inicializace microSD karty

Na Obr. 23 je znázorněn algoritmus inicializace microSD karty, ve kterém jsou využity níže zmíněné funkce. Většina funkcí pracujících s microSD kartou má návratový typ *FRESULT*, což je v podstatě struktura dat s daným významem. Příkladem vrácené hodnoty může být např. *FR\_OK*, *FR\_NOT\_READY* nebo *FR\_NO\_FILE\_SYSTEM*.

- *FRESULT f\_mount(BYTE Drive, FATFS\* FatFileSystem)*

Funkce slouží k inicializaci souborového systému na paměťovém médiu. Inicializace je prováděna na nejnižší úrovni, čili HW inicializace karty a následně i načtení souborového systému. První vstupní proměnná označuje číslo jednotky na microSD kartě, resp. označuje číslo oddílu. Druhým parametrem je ukazatel na datovou strukturu pro uložení údajů z FAT pro patřičný oddíl na paměťové kartě. Funkce vrací výsledek operace:

- *FR\_OK*: Funkce proběhla úspěšně
- *FR\_INVALID\_DRIVE*: Číslo označující oddíl není správné

Touto funkcí je možno médium odpojit a toto je provedeno změnou druhého vstupního parametru na hodnotu *NULL* nebo 0.

- *FRESULT f\_open(FIL\* FileObject, const char\* FileName, ByteModeFlags)*

Tato funkce slouží k otevření souboru v určité konfiguraci upřesněné níže. První vstupní parametr funkce je ukazatel na datovou strukturu, do níž se uloží informace o vybraném souboru. Druhý vstupní parametr je ukazatel na název souboru k otevření a třetím parametrem je struktura parametrů (viz. Tabulka 13).

Typ parametru	Popis
<i>FA_READ</i>	Otevře soubor pouze pro čtení
<i>FA_WRITE</i>	Otevře soubor pouze pro zápis
<i>FA_OPEN_EXISTING</i>	Otevře soubor, pokud neexistuje, funkce se neprovede
<i>FA_OPEN_ALWAYS</i>	Otevře soubor, pokud neexistuje, vytvoří nový soubor
<i>FA_CREATE_NEW</i>	Vytvoří nový soubor, pokud již existuje, funkce se neprovede
<i>FA_CREATE_ALWAYS</i>	Vytvoří vždy nový soubor, existující soubor je nahrazen

Tabulka 13: Parametry souboru funkce *f\_open*

- *FRESULT f\_write(FIL\* FileObject, councst void\* Buffer, WORD ByteToWrite, WORD\* ByteWritten)*

Výše zmíněná funkce slouží k zápisu dat do souboru specifikovaného ukazatelem

*FileObject*. Data k zapsání jsou uložena v bufferu, na který odkazuje druhý vstupní parametr funkce *Buffer*. Třetí parametr označuje počet bytů k zapsání do souboru a čtvrtý parametr označuje již zapsaná data. V podstatě tento parametr určuje pozici v souboru, na kterou se data mají uložit. Pokud je soubor nově otevřen, pozice může být nulová, ale pokud je soubor starý a data se k němu mají připojit, je nutné nejprve zjistit pozice posledních dat. Návrátové hodnoty funkce jsou popsány v Tabulka 14 a jsou shodné s návratovými hodnotami funkce *f\_read*.

- *FRESULT f\_read(FIL\* FileObject, void\* Buffer, WORD ByteToRead, WORD\* ByteRead)*

Funkce pro čtení ze souboru je obdobná funkci pro zápis, pouze se liší druhý vstupní parametr funkce, který ukazuje na pole, do něhož se mají data uložit. Návrátové hodnoty funkce jsou popsány v Tabulka 14.

Hodnota	Význam
FR_OK	Operace úspěšná, hodnota FR_OK = 0
FR_DENIED	Funkce neproběhla, souboru je otevřen pouze pro zápis
FR_RW_ERROR	Funkce neproběhla, možná chyba disku nebo interní chyba
FR_NOT_READY	Funkce neproběhla, chybí médium nebo jiná příčina
FR_INVALID_OBJECT	Objekt souboru není v platném rozsahu

Tabulka 14: Návrátové typy funkcí *f\_read* a *f\_write*

- *FRESULT f\_lseek(FIL\* FileObject, DWORD Offset)*

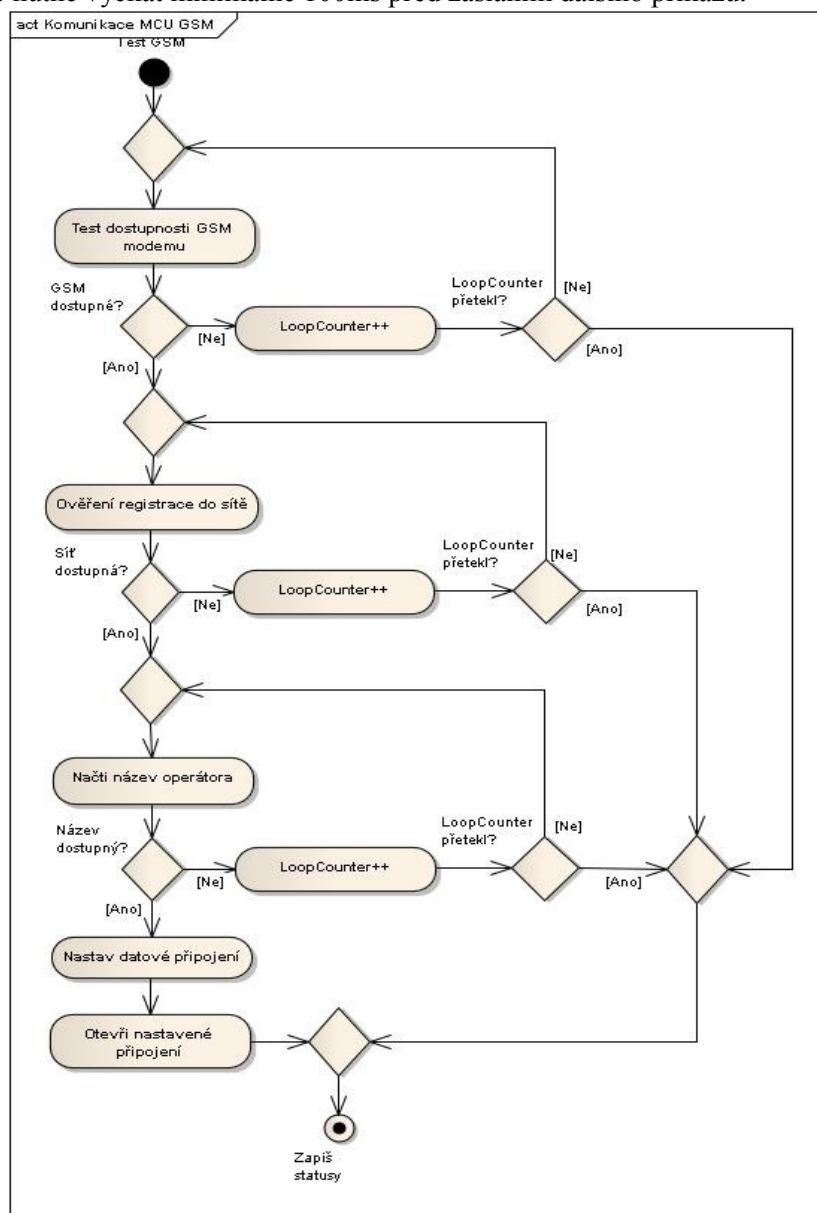
Tato funkce je využita u otevřených souborů k posunutí hodnoty ukazatele o daný offset. Je to nutné při zápisu nebo čtení do existujícího souboru. Nejprve se zjistí velikost souboru a poté se tato hodnota vloží jako druhý vstupní parametr funkce.

- *FRESULT f\_close(FIL\* FileObject)*

Tato funkce slouží k zavření otevřeného souboru specifikovaného vstupní parametrem *FileObject*.

## 6.2.4 Komunikace s GSM modemem

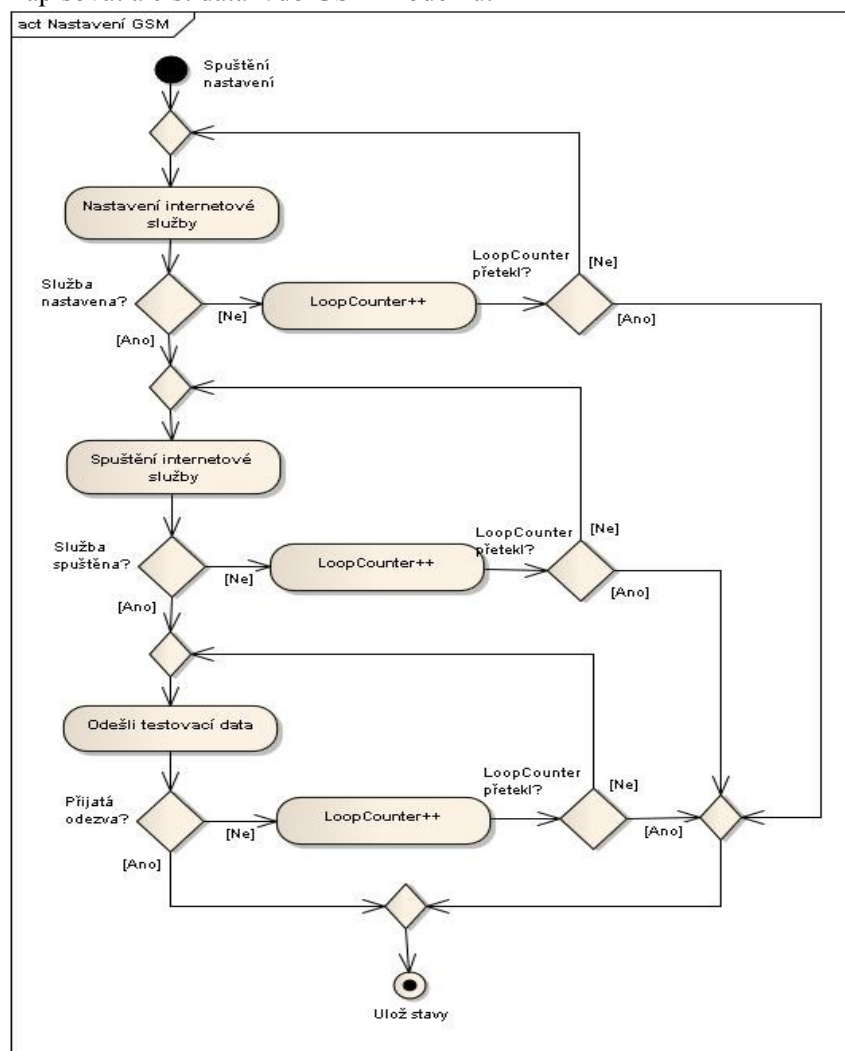
Celá komunikace s GSM probíhá prostřednictvím sady AT příkazů podporovaných modemem. Soubor AT příkazů je dostupný v dokumentaci výrobce[8] GSM modulu v modemu prostřednictvím sběrnice RS232. Na DPS je umístěn ještě obvod MAX3232, což je převodník napěťových úrovní RS232 a UART s napájením 3,3V. Příkaz se vždy skládá z prefixu „at“ nebo „AT“ a odpovídajícímu sufixu pro daný typ příkazu s ukončovacím řetězcem „\r\n“. Na každý zadaný příkaz je po jeho dokončení odeslána odpověď „OK“ nebo „ERROR“. Po obdržení odpovědi je nutné vyčkat minimálně 100ms před zasláním dalšího příkazu.



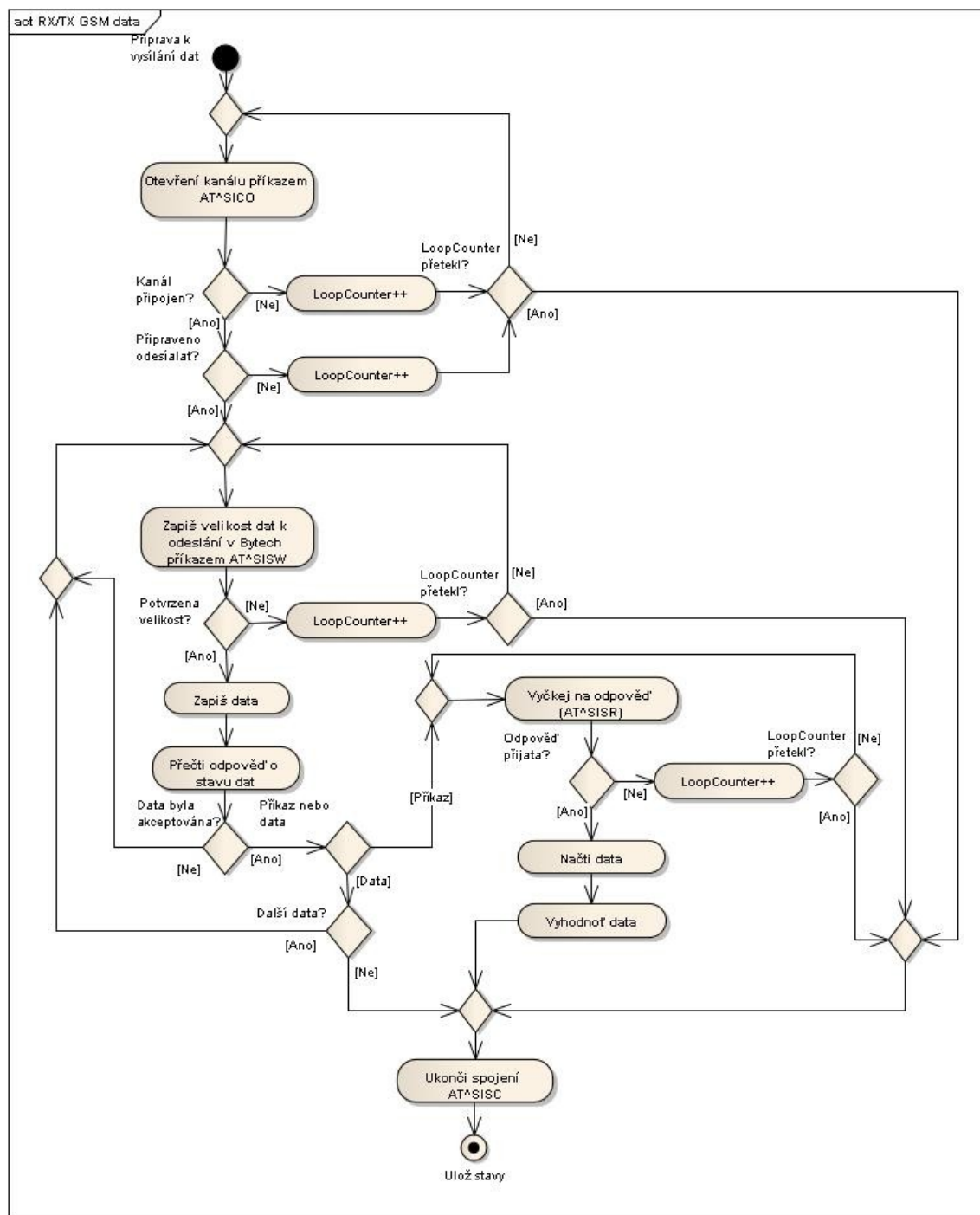
Obr. 24: Diagram aktivit pro přihlášení do GSM sítě

Na Obr. 24 je znázorněn diagram aktivit pro otestování funkce GSM modemu. Úspěšným absolvováním tohoto testu je GSM modem připraven k nastavení připojení k serveru pro příjem a pozdější komunikaci. Tento test je volán periodicky do doby, než je nastavení provedeno.

Dále je zde vytvořen výchozí připojovací profil k připojení na statickou IP adresu a příslušný port. V případě, že je zvolena funkce automatického odesílání dat na server, je v momentě při úspěšně absolvovaném testu okamžitě zkoušeno připojení se na server. Těchto pokusů proběhne jenom omezené množství, kdy se po vyčerpání limitu přestane nadále komunikovat a čeká se na manuální spuštění. Tento profil je možno změnit za pomoci MDA připojeného skrze Bluetooth modul. Nastavení internetového profilu je prováděno dle blokového schématu na Obr. 25. Po úspěšném připojení k serveru a přijetí patřičné odezvy je již možno zapisovat a číst data z/do GSM modemu.



Obr. 25: Nastavení internetové služby GSM



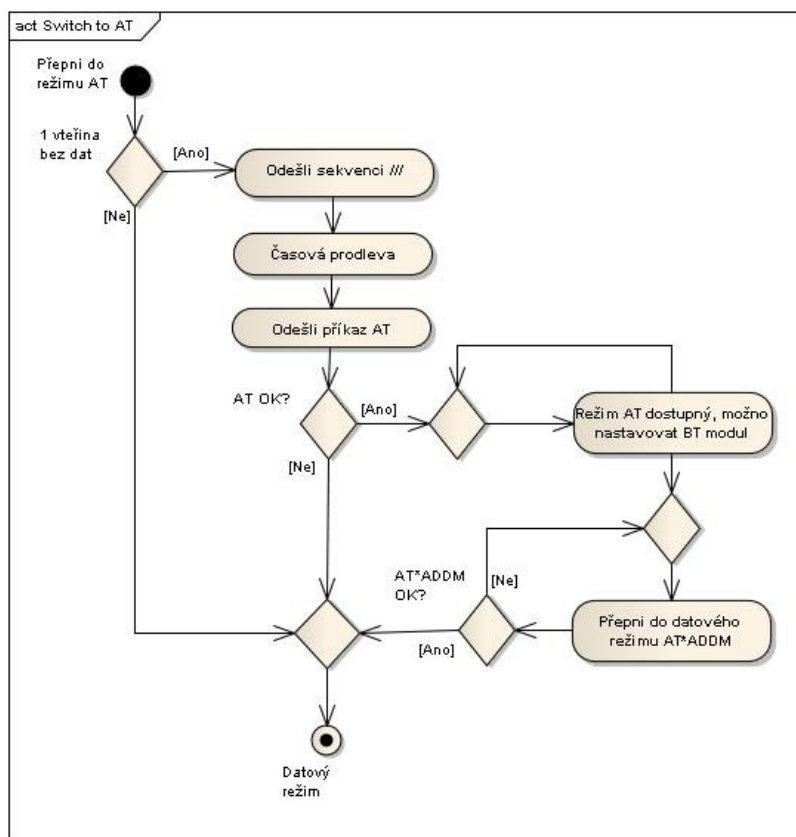
Obr. 26: Komunikace se serverem skrze GSM modem

Pro odeslání dat na server slouží algoritmus dle Obr. 26. Komunikační protokol je navržen tak, že každé odeslání dat je potvrzeno zprávou o přijetí či nepřijetí dat serverem. IP adresa serveru, se kterým se komunikuje je 158.196.72.72 a komunikační port je 12289. Jedná se o soketovou

TCP komunikaci. Adresu serveru a komunikační port lze posléze změnit pomocí programu v MDA.

### 6.2.5 Komunikace s Bluetooth modulem

Při komunikaci s Bluetooth modulem se taktéž využívá AT příkazů, avšak pouze pro ovládání samotného Bluetooth modulu. Bluetooth modul má dostupné dva režimy komunikace, datový a řídicí. V datovém režimu je vytvořen bridge mezi rozhraním UART a Bluetooth, v řídicím režimu je modul možné ovládat pomocí AT příkazů, ostatní data nejsou akceptována a jsou zahozena. Na Obr. 27 je zobrazen algoritmus pro přepnutí do režimu AT. Před odesláním tzv. ESCAPE SEQUENCE (//) nesmí být minimálně 1 vteřinu datový provoz, jinak se tato sekvence považována za přenášená data a následující příkazy taktéž. Při úspěšném vstoupení do režimu AT příkazů je možné konfigurovat nastavení Bluetooth modulu dle našich požadavků.



Obr. 27: Algoritmu přepnutí do AT režimu

Při využití tohoto modulu v mé práci jsem pouze nastavil možnost konfigurace Bluetooth modulu skrze bluetooth připojení, tzn. není nutné připojovat další kabely pro změnu nastavení. Tato změna se provedla prostřednictvím příkazu AT\*ACCB. Seznam platných AT

příkazů je v dokumentaci výrobce [15].

### 6.2.6 Komunikace s GPS modulem

Komunikace s GPS modulem probíhá po komunikační sběrnici UART. V podstatě se jedná pouze o jednosměrnou komunikaci, jelikož GPS modul sám o sobě nepotřebuje nijak inicializovat a periodicky odesílá v daných časových intervalech všechny GPS věty, které byly specifikovány v Tabulka 10. V této práci využívám pouze GPS větu GGA nesoucí nezbytně nutné informace o poloze. Věta GGA může vypadat následovně:

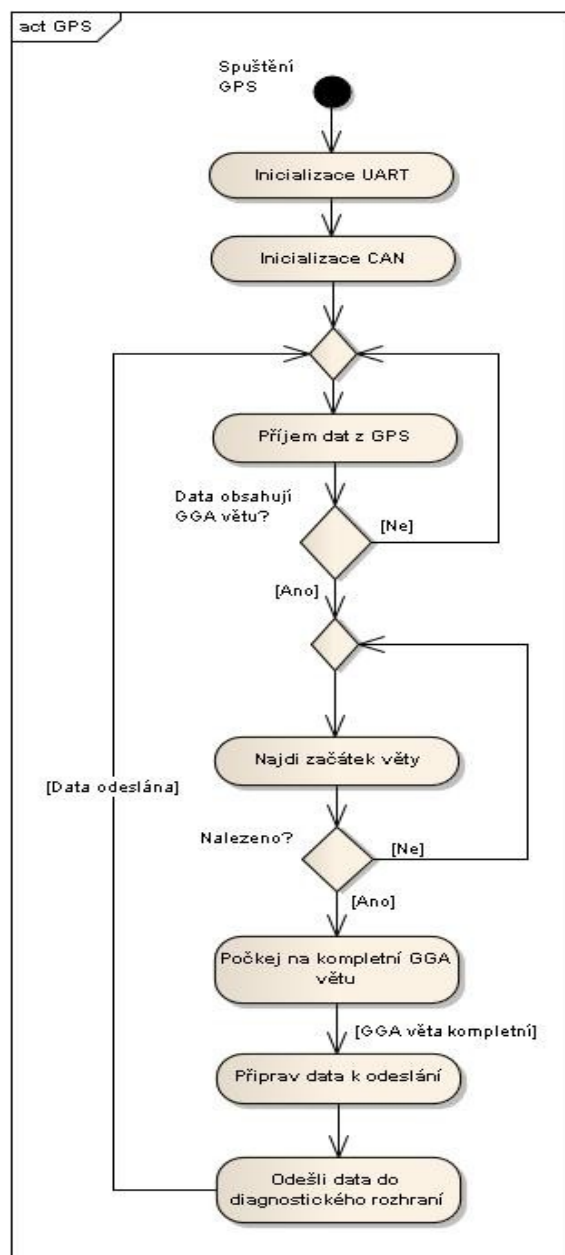
*\$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,\*47*

Prvek	Význam
GGA	GPS pozice fixována
123519	Čas kdy došlo k fixaci
48° 07,038' ,N	Souřadnice polohy, sever nebo jih (N, S)
11° 31',E	Souřadnice polohy, Východ nebo západ (E, W)
1	Kvalita fixace
8	Počet satelitů k trasování
0.9	Horizontální rozptyl pozice
545.5,M	Nadmořská výška
46.9	Výška geoidu (nad hladinou moře)
,,	Prázdné pole
*47	Kontrolní součet, vždy začíná znakem '*'

Tabulka 15: Struktura GGA věty odesílané GPS modulem

Na Obr. 28 je rozkreslený diagram aktivit pro načítání dat z GPS modulu. MCU vybírá pouze zprávy GGA a následně je ukládá do bufferu, který po naplnění odesílá po sběrnici CAN do diagnostického rozhraní. Z MCU načítající data z GPS se po sběrnici CAN odesílá 10 zpráv, které obsahují vždy 8B dat, což je celkem jedna GGA věta. Při načítání se vyčkává na znak '\$', což je symbol označující začátek GPS GGA věty.





Obr. 28: Odeslání GPS věty GGA

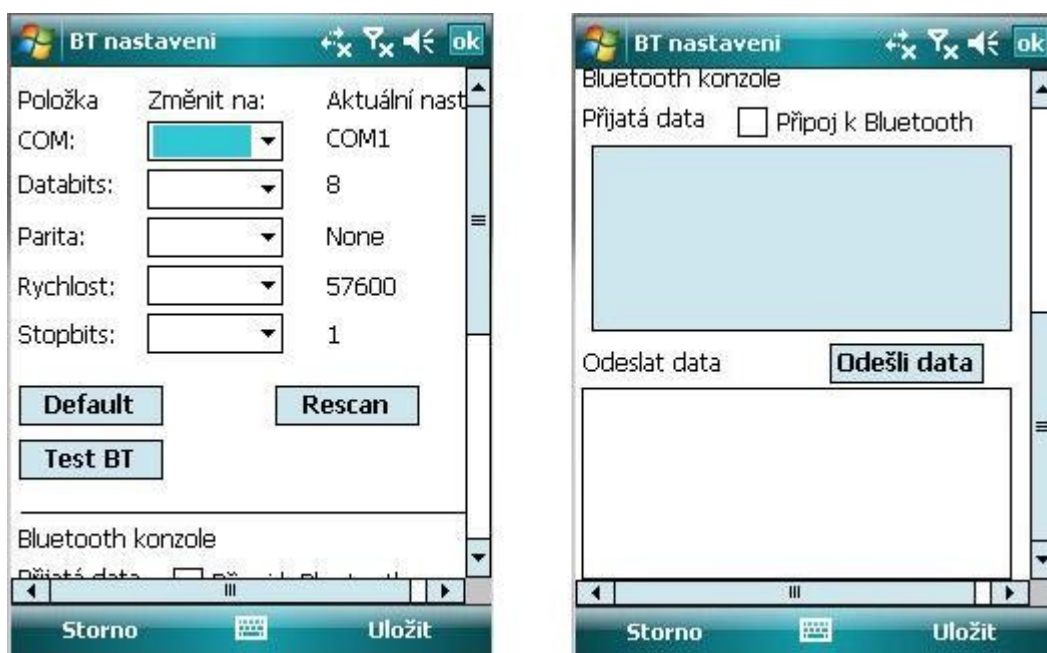
Dekódování GPS dat, přijatých diagnostickým rozhraním na sběrnici CAN, je vytvořeno inverzně k algoritmu odesílání dat. Každá CAN zpráva je testována na přítomnost znaku '\$' a v případě shody, je uloženo 10 zpráv sekvenčně do paměti a následně zapsáno na microSD kartu.

## 7 Návrh aplikace v mobilním zařízení

Aplikace je určená pro mobilní zařízení s operačním systémem Windows Mobile 6.x a její primární funkce je pro nastavení diagnostického rozhraní elektromobilu případně pro nastavení GSM modemu. Dále umožňuje kopírování souborů z microSD karty umístěné v jednotce elektromobilu a případně i jejich smazání. Tyto soubory je pak možné skrze TCP komunikaci odeslat na server. Podmínkou je dostupnost internetového připojení Wi-Fi. Dále je možnost ukládat data do SQL databáze a posléze tyto data přenést na server.

### 7.1 Komunikace s diagnostickým rozhraním

Pro komunikaci s diagnostickou jednotkou je využit virtuální sériový port, který je vytvořen při párování mobilního zařízení s Bluetooth modulem. Po přidělení čísla sériového portu je nutné v aplikaci nastavit parametry použitého COM portu. Na Obr. 29 je vyobrazena obrazovka z MDA pro nastavení parametrů COM portu. V levé části obrázku se nastavují parametry a v pravé části je možno komunikaci otestovat.



Obr. 29: Nastavení COM portu v MDA

Po úspěšném nastavení parametrů komunikačního kanálu je při stisku položky menu *Uložit* celá konfigurace nastavení portu uložena do registrů a s každým spuštěním aplikace se toto nastavení načítá. Při spouštění aplikace dochází k pokusu otevření připojení s k Bluetooth

modulu a následně k otevření portu. Jestliže je tato operace neúspěšná, je uživatel upozorněn chybovým hlášením.

## 7.2 Komunikační protokol diagnostického rozhraní

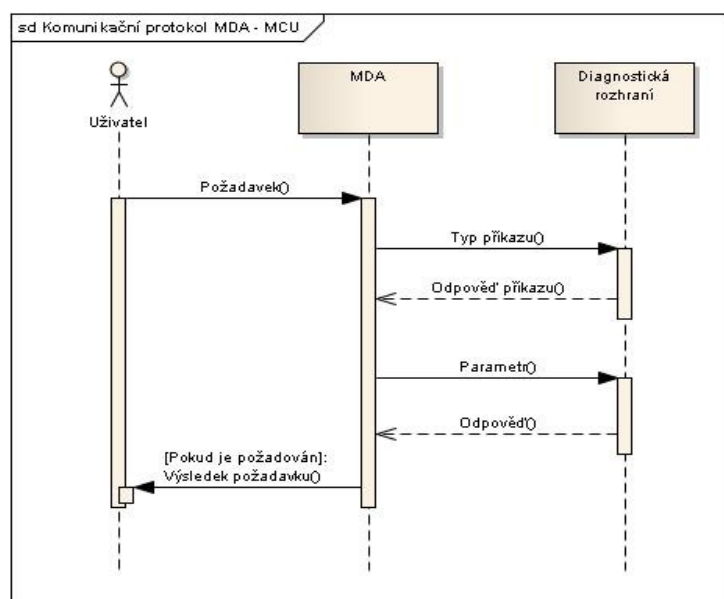
Při vytváření aplikace byly využity poznatky z komunikace pomocí AT příkazů a vznikl komunikační protokol založený na řízené komunikaci za pomoci příkazů a odpovědí. Každý příkaz se skládá z prefixu CMD a následně čísla, označující typ příkazu. Dále následuje ukončovací znak '\n', který značí nový řádek. Ve výsledku může příkaz vypadat takto: „CMD1\n“. Po obdržení příkazu je odeslána odpověď zpět do MDA, a to buď „OK\n“ nebo „ERROR\n“. Obě odpovědi jsou taktéž ukončeny znakem nového řádku. Pokud není obdržena odpověď v platném časovém intervalu, je uživatel o této skutečnosti informován. V Tabulka 16 je přehled využívaných příkazů pro komunikaci mezi MDA a MCU v diagnostickém rozhraní a v Tabulka 17 je popis příkazů, u kterých je využit navíc i parametr. Na Obr. 30 je znázorněn sekvenční diagram komunikace příkazu s parametrem. Při odeslání příkazu bez parametru je zde jediný rozdíl, a to že po odeslání příkazu nenásleduje po odpovědi odeslání parametru.

Příkaz	Popis příkazu
CMD1	Odešle soubor s daty ze systémové sběrnice CAN1
CMD2	Zapnutí nebo vypnutí ukládání dat z GPS na sběrnici CAN2
CMD3	Zapíná nebo vypíná automatické odeislání dat z externí GPS do MDA
CMD4	Změna rychlosti sběrnice UART mezi MCU a Bluetooth modulu
CMD5	Zapíná bridge mezi sběrnici UART z Bluetooth modulu a UART pro GSM modem
CMD6	Odeslání stavu GSM modemu do MDA
CMD7	Smazání souboru na microSD kartě
CMD8	Nastavení adresy a portu serveru

*Tabulka 16: Přehled příkazů pro diagnostické rozhraní od MDA*

Příkaz	Parametr
CMD1	Název souboru k přenosu včetně přípony
CMD4	Požadovaná rychlost komunikační sběrnice
CMD7	Kompletní název souboru k smazání
CMD8	Kompletní IP adresa serveru : číslo portu (např. 192.168.1.1:12289)

Tabulka 17: Parametry některých příkazů



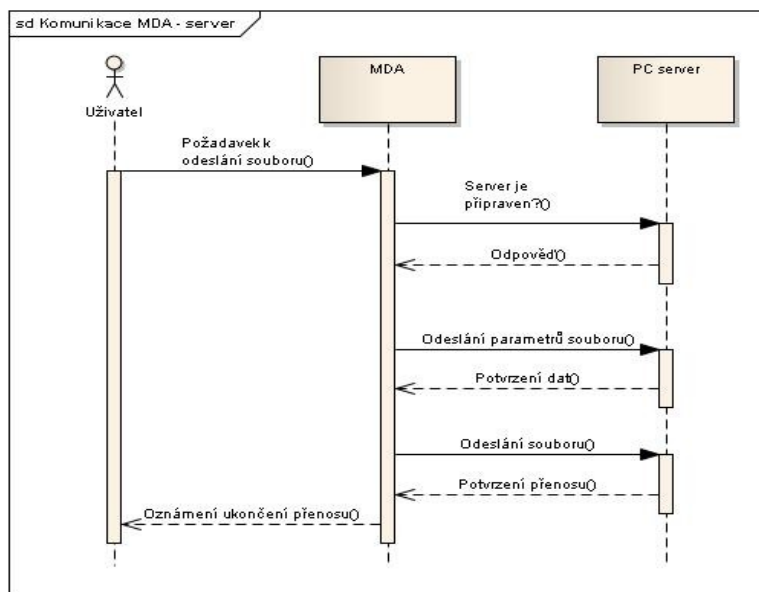
Obr. 30: Sekvenční diagram komunikace

### 7.3 Komunikace mezi MDA a serverem

Komunikace mezi MDA a serverem je uskutečněna prostřednictvím TCP protokolu. Je využita TCP komunikace implementovaná v rozhraní .NET Compact Framework. Komunikace slouží pouze k odeslání souboru z MDA, které nejprve načte požadovaný soubor z diagnostického rozhraní. Žádné jiné využití této komunikace v této práci není.

Komunikační kanál je za chodu programu uzavřen a otevírá se až při požadavku uživatele k odeslání souboru na server. Tento princip přenosu souboru je vytvořen z důvodu minimalizování nutnosti manipulace s kabeláží. Při zadání požadavku o odeslání souboru, se zahájí otevírání komunikačního kanálu a při úspěšném otevření se informuje server o požadavku odeslat soubor. V případě, že je server zaneprázdněn, odešle do MDA informaci o tomto stavu a uživateli se zobrazí informativní hlášení. Pokud bude server připraven soubor přijmout, odešle odpověď značící připravenost k příjmu. MDA reaguje na tuto odpověď

odesláním informací o souboru, tj. název souboru a jeho velikost. Server tuto informaci dekóduje, vytvoří soubor s daným jménem na diskovém prostoru PC a odešle MDA informaci s povolením k odeslání souboru. V případě, že server nedokáže dekódovat zprávu, odešle tuto informaci do MDA a to se pokusí odeslat vždy tu zprávu, u které byl tento požadavek vystaven. Tento algoritmus je popsán na Obr. 31, ovšem je zde popsán pouze případ komunikace bez výskytu chyb.



Obr. 31: Sekvenční diagram komunikace MDA - server

Nastavení serveru

Přidělené IP adresy: 127.0.0.1

IP adresa serveru: 158.196.72.72

Port: 12291

Test spojení

Odešli testovací soubor

Zpět Nastavení

*Obr. 32: Nastavení serveru v MDA*

Pro nastavení parametrů serveru, jako je IP a port je zobrazena obrazovka na Obr. 32. Přednastavená konfigurace je určena pro spojení na server a primárně slouží k jednodušší obsluze. Samozřejmě lze nastavit vlastní adresu i port a po kliknutí na položku menu *Nastavení* je možné tuto konfiguraci uložit do registrů.

## 7.4 Komunikační protokol serveru

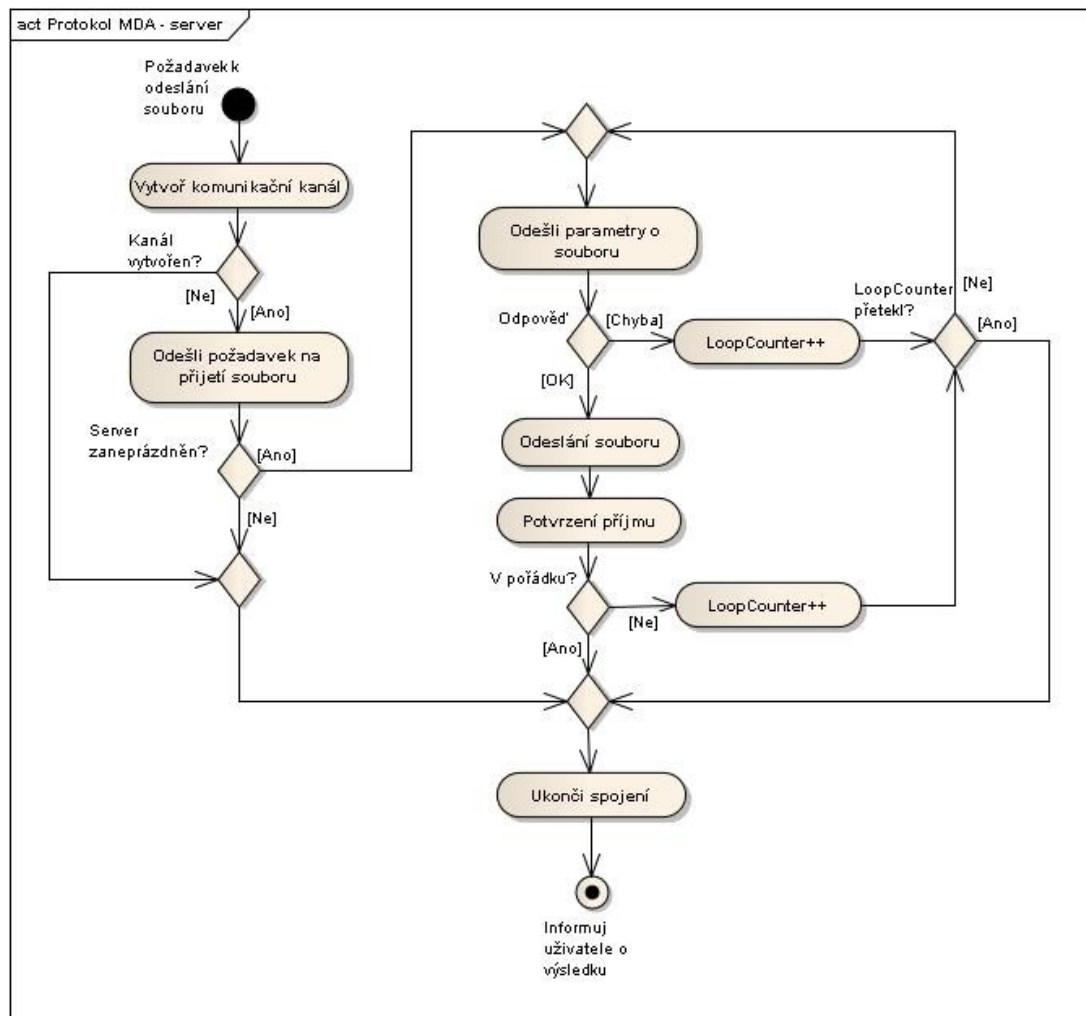
Komunikace mezi MDA a serverem je uskutečněna pomocí TCP protokolu. Pro lepší zapamatování nastavení komunikačního kanálu v MDA je IP adresa serveru stejná jak pro nastavení MCU, tak i pro MDA. Odlišnost je však v komunikačním portu. Nastavený port pro komunikační kanál mezi diagnostickou jednotkou a serverem je 12289 a komunikační port pro komunikační kanál MDA a server je 12291.

Na Obr. 33 je pak podrobněji popsán rozbor komunikačního protokolu pro komunikaci mezi mobilním zařízením a serverem. Během zpracovávání požadavku je dostupná lokální proměnná, do níž se zapisují stavy, ve kterých se proces nachází. Dle této proměnné je po ukončení komunikace uživatel informován o výsledku operace. Pokud dojde v určitém stavu k selhání, uloží se do této proměnné hodnota odpovídající nastalé chybě. Pokud však komunikace proběhne bez jakýchkoliv problémů, je hodnota této proměnné nulová a uživateli se zobrazí informační dialog o úspěšném provedení operace.

Odpověď odeslaná serverem má dvě varianty, viz. Tabulka 18.

Odpověď	Popis odpovědi
OK	Nevyskytla se žádná chyba, operace úspěšná
ERROR	Vysytl se chyba, operace selhala

Tabulka 18: Odpověď ze strany serveru



Obr. 33: Vývojový diagram komunikačního protokolu pro MDA - server

Jednotlivé příkazy a odpovědi jsou ukončovány znakem nového řádku '\n'. Tímto rozlišením je možno navázat zpětnou komunikaci skrze komunikační kanál a také je tímto způsobem ošetřena variabilní délka požadavku k odeslání dat.

## 8 Návrh aplikace na PC – server

Tato aplikace je především určená pro PC s operačním systémem Windows XP s .NET Framework 3.5. Podmínkou funkčnosti komunikace mezi diagnostickým rozhraním v elektromobilu je, aby měl PC s touto aplikací přidělenou veřejnou IP adresu a neblokovaný TCP port 12 289. Veřejná IP adresa je nutná kvůli dostupnosti serveru z internetu, protože pokud by se jednalo o neveřejnou IP adresu, která je např. ještě za několika routery či proxy servery, je v podstatě nemožné se na tento server připojit. Určitou možnost by zde nabízel tzv. „Port Triggering“, ale s touto možností jsem u svého poskytovatele neuspěl. Lepší a hlavně jistější verzí řešení je připlatit si u svého poskytovatele internetu určitou částku a nechat si přidělit veřejnou IP adresu, pokud to ovšem poskytovatel umožňuje. Na druhou stranu, je zde i nevýhoda spjatá s veřejnou IP adresou a to, že je potom PC hůře chráněno před útoky z internetu. Tento negativní faktor lze snížit použitím routeru s funkcí NAT, čili překladem adres, který skryje počítače v síti a tím omezí míru rizika.

Aplikace je navržena spíše pro sběr dat z MDA nebo diagnostického rozhraní a jejich následnému prohlížení. V aplikaci běží dva oddělené procesy, které naslouchají na daných nastavených portech a všech dostupných připojených síťových prostředků. Vlákno, které naslouchá na portu 12288 má za úkol přijímat veškerá data a ukládat do SQL databáze. Ukládání dat probíhá po naplnění fronty dat a při naplnění tohoto bufferu jsou data zapsána do databáze. V případě, že se spojení přeruší, jsou data zapsány okamžitě. V hlavním vlákne aplikace je pak možnost prohlížet data v SQL databázi. Druhé vlákno je určeno pro komunikaci s mobilním zařízením, resp. pro příjem souboru.

### 8.1 Komunikace s diagnostickým rozhraním

Komunikace s diagnostickým rozhraním může probíhat obousměrně, avšak spojení musí vždy zahájit jednotka diagnostického rozhraní, jelikož GSM modem nemá přidělenou IP adresu a je v podstatě nedostupný. Po zahájení komunikace je komunikační kanál zachován po celou dobu spuštění diagnostického rozhraní. Během otevřeného komunikačního kanálu lze zapisovat do *NetworkStream* data, která jsou následně odeslána na GSM modem. Ten informuje MCU o přijatých datech, které jsou dostupné v FIFO bufferu modemu. TCP server naslouchá na všech dostupných síťových rozhraních a komunikační port je vybrán 12288. V této práci se využívá pouze jednosměrné komunikace a to z GSM modemu na server. Odesílají se celé vybrané zprávy ze sběrnice CAN, tzn. včetně ID a masky zprávy společně s daty. Tyto data jsou ukládána a postupně zapisována do SQL databáze v serveru společně s časovým razítkem.



### 8.1.1 Komunikační protokol

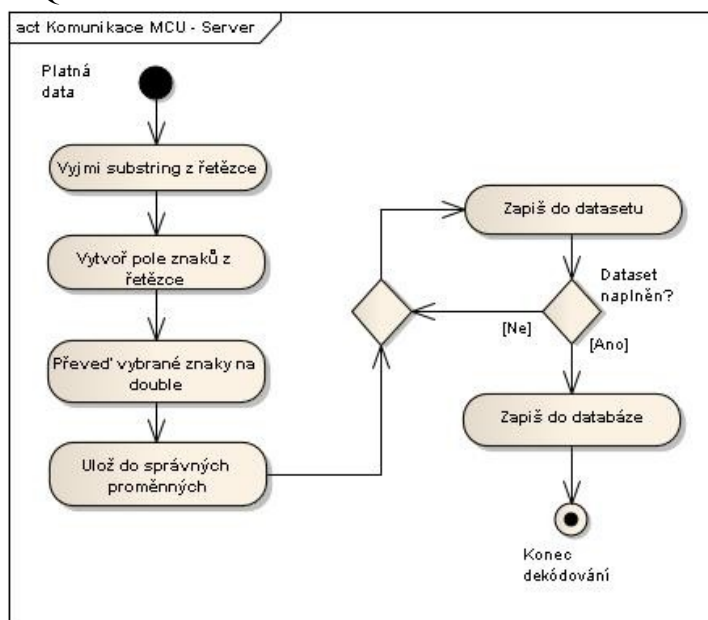
Komunikační protokol je kvůli minimalizaci přenosu dat a tím i zvýšení komunikační rychlosti ve smyslu zvětšení míry užitečných dat takřka prázdný. K největšímu zpoždění dochází při čtení dat z GSM modemu, kdy je nutné dodržovat prodlevu při zadávání AT příkazů, tudíž nedochází k potvrzování dat serverem, ale spoléhá se pouze na mechanismus protokolu TCP. Data, která jsou určená pro odeslání na server jsou v diagnostickém rozhraní seřazena do pole znaků o velikosti 512B, což je ve skutečnosti zásobník na 32 zpráv a nakonec je přidán znak '\n' informující o konci dat. Je to vytvořeno z důvodu úspory dat ve formě hlaviček a dalších informací, které odesílá GSM modem spolu s daty. Struktura 1 zprávy je v Tabulka 19.

Hlavička[5B]	Obsah zprávy [512B]	Ukončovací znak [1B]
"DATA:"	Obsah bufferu MCU	'\n'

Tabulka 19: Struktura komunikačního protokolu z Diagnostického rozhraní

### 8.1.2 Příjem a dekodování dat

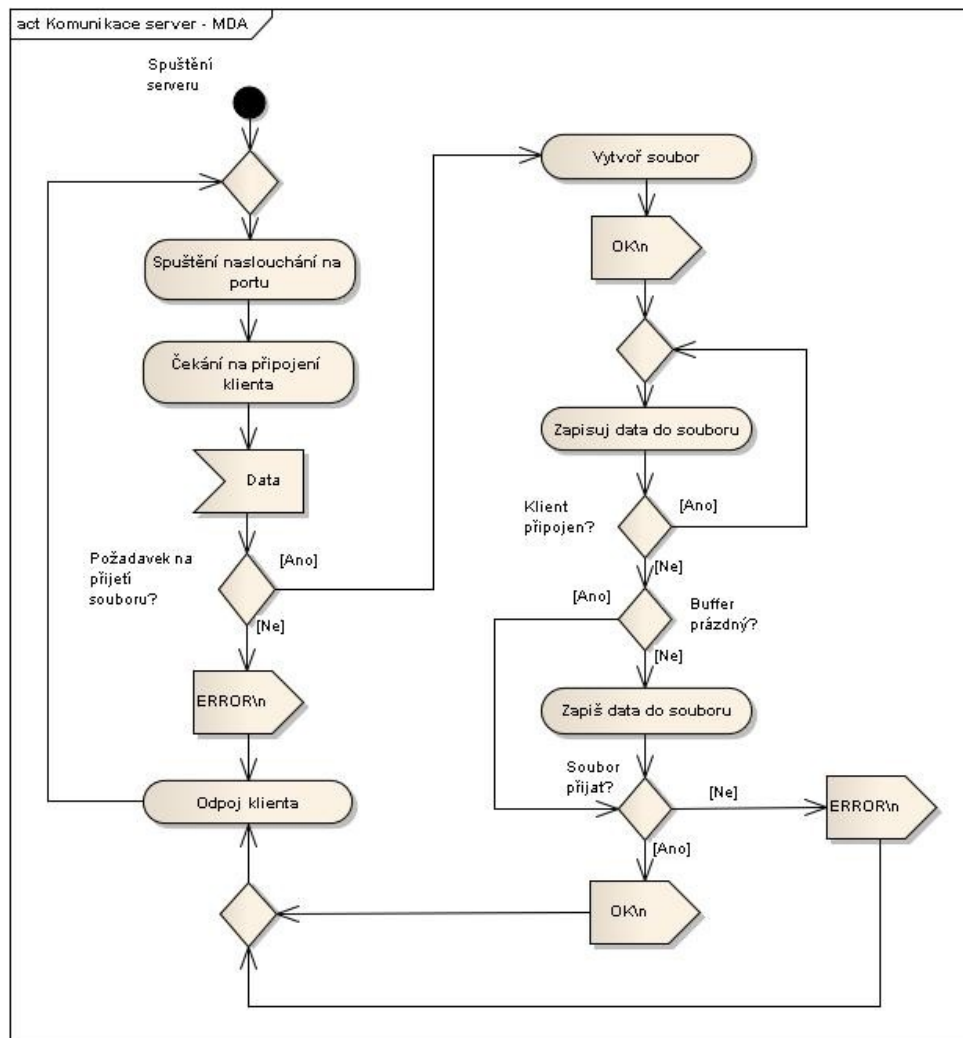
Jelikož je známá velikost přenášených dat, je možno po dokončení příjmu tyto data zkontrolovat porovnáním velikosti přijatého řetězce. Pakliže velikost nesouhlasí, jsou tyto data nežádoucí a jsou zahozena. Pokud ale řetězec kontrolou projde, je z něj odebrán subřetězec a ten je převeden na pole znaků o velikosti 512B. Jednotlivé znaky jsou pak tříděny do patřičného formátu a uloženy do SQL databáze.



Obr. 34: Vývojový diagram dekodování a ukládání dat

## 8.2 Komunikace s mobilní aplikací

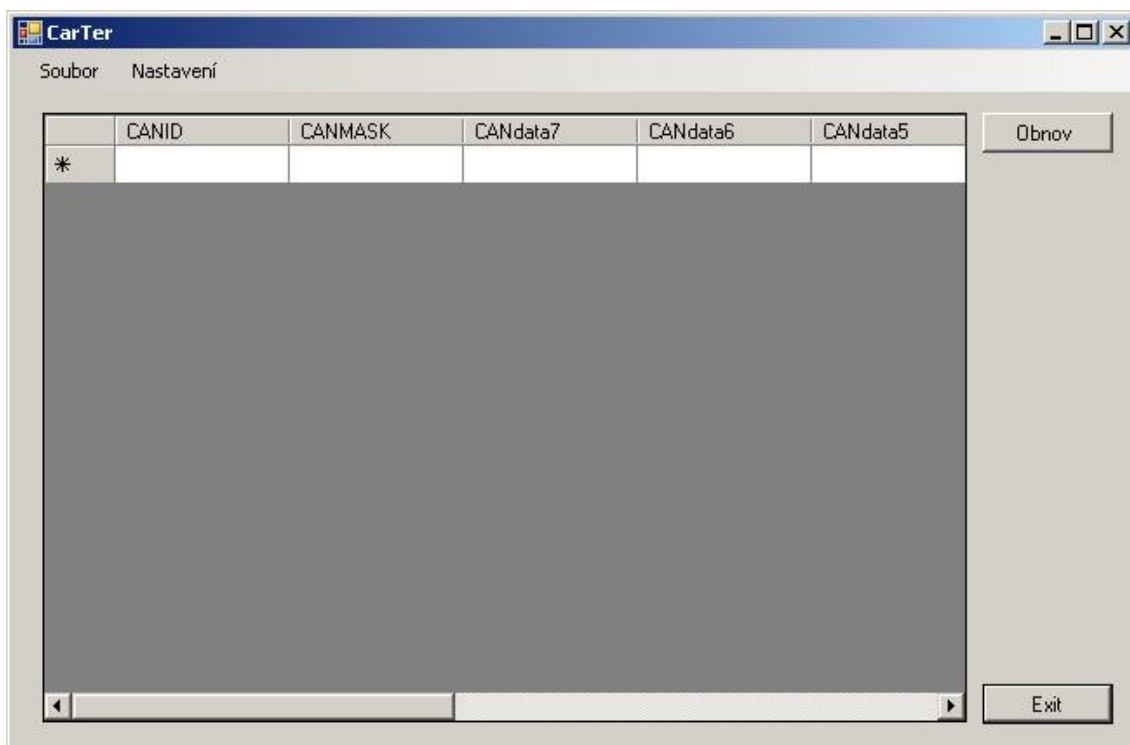
Komunikace s mobilní aplikací je zahájena připojením mobilního zařízení k serveru. Na Obr. 35 je znázorněn vývojový diagram rutiny TCP serveru. Tato rutina běží ve vlastním vlákne, protože čekáním na připojení klienta se vlákno zablokuje, což znamená, že pokud by tato procedura běžela ve stejném vlákne jako celá aplikace, došlo by k zablokování celé aplikace.



Obr. 35: Vývojový diagram TCP serveru

Po akceptování požadavku připojení klienta se automaticky čeká na data, ukončená znakem '\n', což značí, že je dostupný požadavek. Jakmile jsou data přijata, testují se na přítomnost validního požadavku. V případě, že data neobsahují správný řetězec, je odeslána zpět do GSM modemu odpověď „ERROR\n“ a klient bude odpojen. Následně se opět čeká

na připojení klienta a na přijetí požadavku. V případě, že požadavek bude dekodován, vytvoří se soubor se jménem přenášeného souboru. Po přijetí celého souboru je nutné odeslat do MDA potvrzení o přijetí souboru a výsledku celé operace, což v úspěšném dokončení operace způsobí odpojení klienta od serveru a upozorní uživatele na tuto skutečnost. Na Obr. 36 je obrazovka demonstrační aplikace na PC. Je primárně určena pro zobrazování databázového souboru, do něhož se ukládají data odesílaná GSM modemem. Je zároveň vytvořena jako server pro příjem souborů z MDA skrze TCP komunikaci.



Obr. 36: Demonstrační aplikace na PC

## 9 Závěr

Cílem této diplomové práce bylo navrhnout vestavné diagnostické rozhraní elektromobilu a využít dostupné komunikační technologie a mobilní zařízení. Navržené diagnostické rozhraní je v Příloha 3. Diagnostické rozhraní úspěšně odposlouchávalo provoz na obou sběrnících CAN a ukládalo data na microSD kartu do souboru. Taktéž se podařilo komunikovat se serverem skrze GSM modem, což byla nejnáročnější část práce z důvodu velkého objemu porovnávaných dat přijatých z modemu. Vzhledem k nemožnosti využít proměnné *string*, tak nešlo využít funkcí jako je např. *strcmp()*. Tento problém byl však posléze vyřešen využitím porovnávání paměti pomocí funkce *memcmp()*. Dále bylo zjištěno, že operace s FAT na paměťové kartě úzce souvisí s velikostí clusteru, jelikož při nesprávném naformátování microSD karty se objevily problémy se souborovými strukturami. Při vytváření aplikace na MDA jsem se setkal s nedostatečným výkonem procesoru, což mělo za následek špatně se zobrazujících ovládacích prvků, např. při přijímání souboru skrze Bluetooth. Tento problém se podařilo vyřešit použitím uspáním aktuálního vlákna na krátkou dobu v řádu desítek až stovek milisekund.

Další vývoj vestavného diagnostického rozhraní by mohl směřovat k zaměření na univerzálnost použití ve více typech dopravních prostředků. Toto by zřejmě znamenalo rozšířit komunikační protokoly s možností variabilních rychlostí komunikačních sběrnic, případně čerpání dat z nějaké znalostní databáze s parametry nastavení pro jednotlivé značky automobilů. Při neustálém vývoji technologií by bylo lepší nahradit dosavadní Bluetooth modul specifikace 2.1 za 3.0, případně i novější a taktéž nahradit GPS modul za modul s větší citlivostí, případně modulem vybaveným novými technologiemi navigačního systému (GALILEO, GLONASS). Další vývoj serverové aplikace by mohl směřovat na grafické vyhodnocení dat, případně by se mohl vytvořit globální webový server s uživatelskou databází pro minimalizaci nákladů na provoz serveru u uživatele. Tímto řešením by se taktéž docílilo možnosti zvětšení zabezpečení automobilu proti odcizení, jelikož by data o pozici byla okamžitě dostupná odkudkoliv.

Diagnostické rozhraní bude mít přínos v oblasti diagnostiky poruch zařízení na sběrnici, jelikož by se mohl zjistit i okamžik, ve kterém se chyba projevila. Ovšem za předpokladu, že je uživateli znám komunikační protokol provozovaného vozidla.

## Seznam literatury

- [1] Elektromobil. [online]. [cit. 2012-01-10]. Dostupné z: <http://cs.wikipedia.org/wiki/Elektromobil>
- [2] NiMH Nickel Metal Hydride Battery Technology. [online]. [cit. 2012-01-10]. Dostupné z: <http://www.radio-electronics.com/info/power-management/battery-technology/nimh-nickel-metal-hydride.php>
- [3] Lithium-based Batteries. [online]. [cit. 2012-01-10]. Dostupné z: [http://batteryuniversity.com/learn/article/lithium\\_based\\_batteries](http://batteryuniversity.com/learn/article/lithium_based_batteries)
- [4] LiFePO4 akumulátory. [online]. [cit. 2012-01-10]. Dostupné z: <http://elektromobil.vseznamu.cz/baterie-v-elektromobilech/lifepo4-akumulatory>
- [5] ČVUT PRAHA. *Komunikační protokol CANopen: Základní informace a zadání úloh* [online]. [cit. 2012-01-10]. Dostupné z: [http://dce.felk.cvut.cz/drs/cviceni/can/doc/popis\\_canopen.pdf](http://dce.felk.cvut.cz/drs/cviceni/can/doc/popis_canopen.pdf)
- [6] ROMÁNEK, David. Co je CANopen a jak na něj. [online]. 2006 [cit. 2012-01-16]. Dostupné z: <http://www.hw.cz/Produkty/Obecne-produkty/ART1563-Co-je-CANopen-a-jak-na-nej.html>
- [7] CIA. *CANOpen* [pdf]. [cit. 2012-01-10].
- [8] CINTERION. *MC75i AT Command Set* [online]. 01.100. 2008 [cit. 2012-04-24].
- [9] SD GROUP. *SD Specifications Part 1: Physical Layer Simplified Specification* [online]. 3.01. San Ramon, CA 94583 USA: SD Card Association, 2010 [cit. 2012-04-24]. Dostupné z: [https://www.sdcard.org/downloads/pls/simplified\\_specs/Part\\_1\\_Physical\\_Layer\\_Simplified\\_Specification\\_Ver\\_3.01\\_Final\\_100518.pdf](https://www.sdcard.org/downloads/pls/simplified_specs/Part_1_Physical_Layer_Simplified_Specification_Ver_3.01_Final_100518.pdf)
- [10] TEXAS INSTRUMENTS. *LM2672 SIMPLE SWITCHER®: Power Converter High Efficiency 1A Step-Down Voltage Regulator with Features* [online]. Texas, 2007 [cit. 2012-04-24]. Dostupné z: <http://www.ti.com/lit/ds/symlink/lm2672.pdf>
- [11] TEXAS INSTRUMENTS. *Stellaris® LM3S2965 Microcontroller* [online]. Austin Texas, 2011 [cit. 2012-04-24]. Dostupné z: <http://www.ti.com/lit/ds/spms062g/spms062g.pdf>
- [12] CONNECTBLUE. *OEM Serial Port Adapter™: Electrical & Mechanical Datasheet* [online]. 1.11. Švédsko, 2007 [cit. 2012-04-24]. Dostupné z: [http://www.spezial.cz/pdf/em\\_ds\\_oemspa\\_311\\_331.pdf](http://www.spezial.cz/pdf/em_ds_oemspa_311_331.pdf)
- [13] SANDISK CORPORATION. *SanDisk Secure Digital Card: Product Manual* [online]. Version 1.9. Sunnyvale, CA, 2003 [cit. 2012-04-30]. 80-13-00169. Dostupné z: <http://www.circlemud.org/~jelson/sdcard/SDCardStandardv1.9.pdf>
- [14] OLMR, Vít. HW server představuje - Sériová linka RS-232. HW SERVER S R.O. *Hw.cz* [online]. 2005, 12. prosinec [cit. 2012-04-30]. Dostupné z: <http://www.hw.cz/rozhrani/hw-server-predstavuje-seriova-linka-rs-232.html>

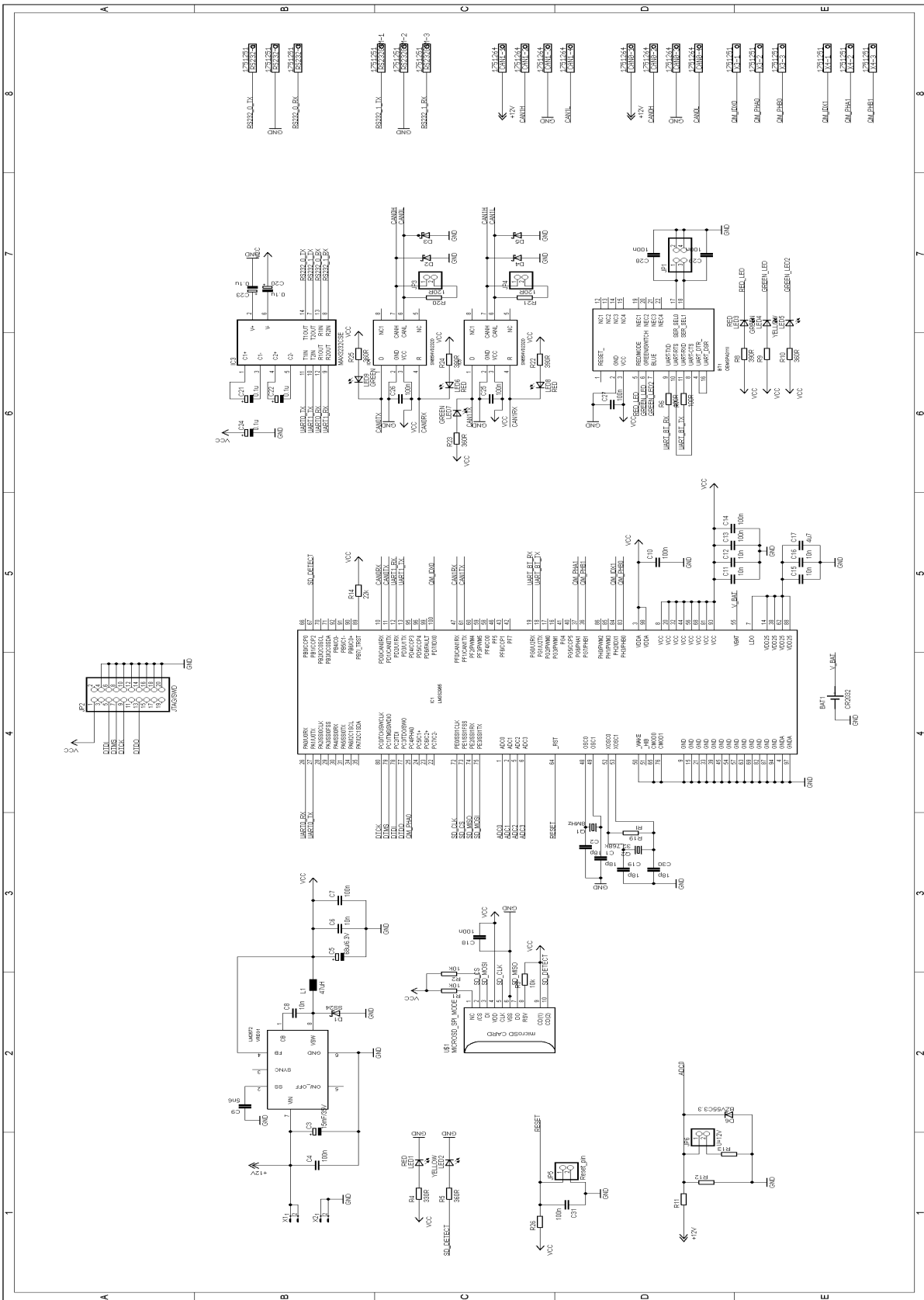
- [15] CONNECTBLUE AB. *BLUETOOTH SERIAL PORT ADAPTER AT COMMANDS* [online]. 18. vyd. 2012 [cit. 2012-05-01]. Dostupné z: [http://www.spezial.cz/pdf/Bluetooth\\_Serial\\_Port\\_Adapter\\_AT\\_Commands.pdf](http://www.spezial.cz/pdf/Bluetooth_Serial_Port_Adapter_AT_Commands.pdf)
- [16] CAN - popis struktury. HW SERVER S R.O. *CAN - popis struktury | HW.cz* [online]. 2004 [cit. 2012-05-01]. Dostupné z: <http://www.hw.cz/navrh-obvodu/rozhrani/can-popis-struktury.html>
- [17] BOSCH. *CAN Specification: Version 2.0* [HDD]. 2.0. Stuttgart: Robert Bosch GmbH, 1991 [cit. 2012-05-01].
- [18] CIA. *CANOpen* [HDD]. Erlangen [cit. 2012-05-01].
- [19] TEXAS INSTRUMENTS. *USER'S GUIDE: Stellaris® Peripheral Driver Library* [HDD]. 8264. vyd. Austin, Texas, 2011, 4. listopad 2011 [cit. 2012-05-03]. Dostupné z: <http://www.ti.com/mcu/docs/mcuorphantoolsw.tsp?sectionId=632&orphantabId=8>
- [20] LEADTEK RESEARCH INC. *LR9552: GPS Module* [online]. V0.4. Taiwan, 2007 [cit. 2012-05-01]. Dostupné z: <http://www.sectron.cz/produkty/31-vyprodej--leadtek-research/32-gps-oem-moduly/655-gps-lr9552-rs232-modul-leadtek.html>
- [21] What is WAAS?. GARMIN. *Garmin | What is WAAS?* [online]. [cit. 2012-05-01]. Dostupné z: <http://www8.garmin.com/aboutGPS/waas.html>
- [22] SECTRON S R.O. *GSM AUTOMATIC: MC75i EDGE TERMINAL* [online]. [cit. 2012-05-04]. Dostupné z: <http://www.sectron.cz/produkty/12-gsm-modem-umts-modem/238-gsm-modem/1156-mc75i-edge-terminal.html>
- [23] Reading an SD card with an ATMEGA168. *Elastic Sheep* [online]. 2010 [cit. 2012-05-01]. Dostupné z: <http://elasticsheep.com/2010/01/reading-an-sd-card-with-an-atmega168/>
- [24] WIGLEY, Andy, Daniel MOTH a Peter FOOT. *Microsoft mobile development handbook*. Redmond: Microsoft Press, 2007, 651 s. ISBN 978-0-7356-2358-3.
- [25] SHARP, John. *Microsoft Visual C# 2005: krok za krokem*. Vyd. 1. Brno: Computer Press, 2006. ISBN 80-251-1156-3.
- [26] TEXAS INSTRUMENTS. *Diagnosing Software Faults in Stellaris Microcontrollers* [HDD]. Preliminary. 2012 [cit. 2012-05-01].

## **Seznam příloh**

Příloha 1	Schéma zapojení Diagnostického rozhraní elektromobilu
Příloha 2	Osazovací plán Diagnostického rozhraní elektromobilu
Příloha 3	Vyrobená DPS Diagnostického rozhraní elektromobilu
Příloha 4	Porovnání velikosti GPS modulu a baterie typu AA
Příloha 5	Připojení GPS modulu k vývojovému kitu CAN

## Příloha I

## Schéma zapojení Diagnostického rozhraní elektromobilu

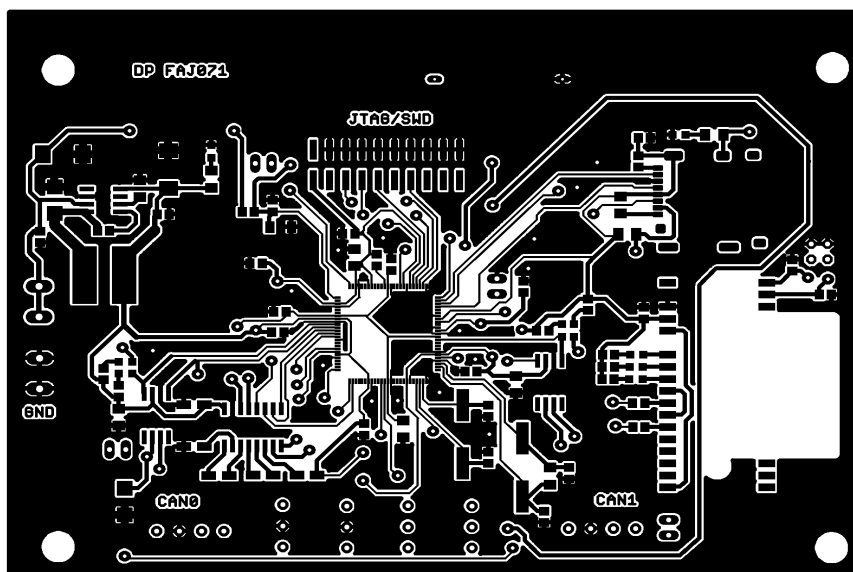




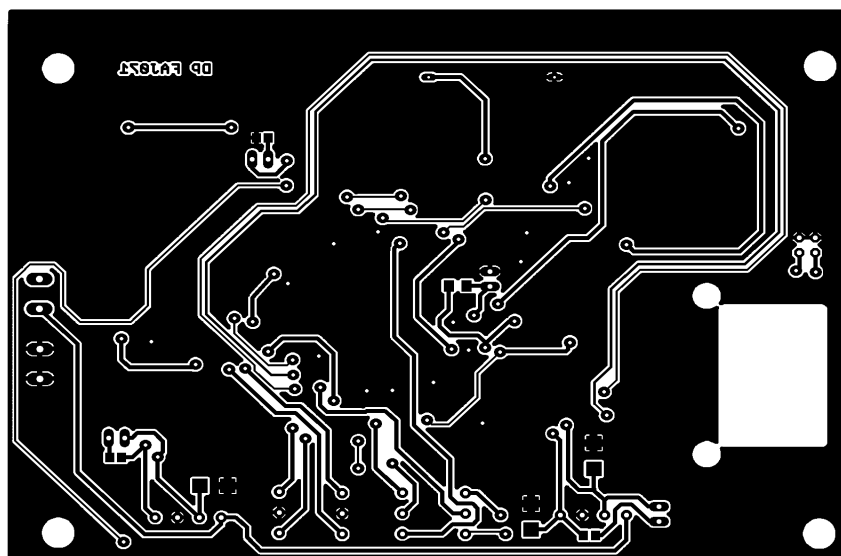
## Příloha II

Osazovací plán DPS diagnostického rozhraní elektromobilu

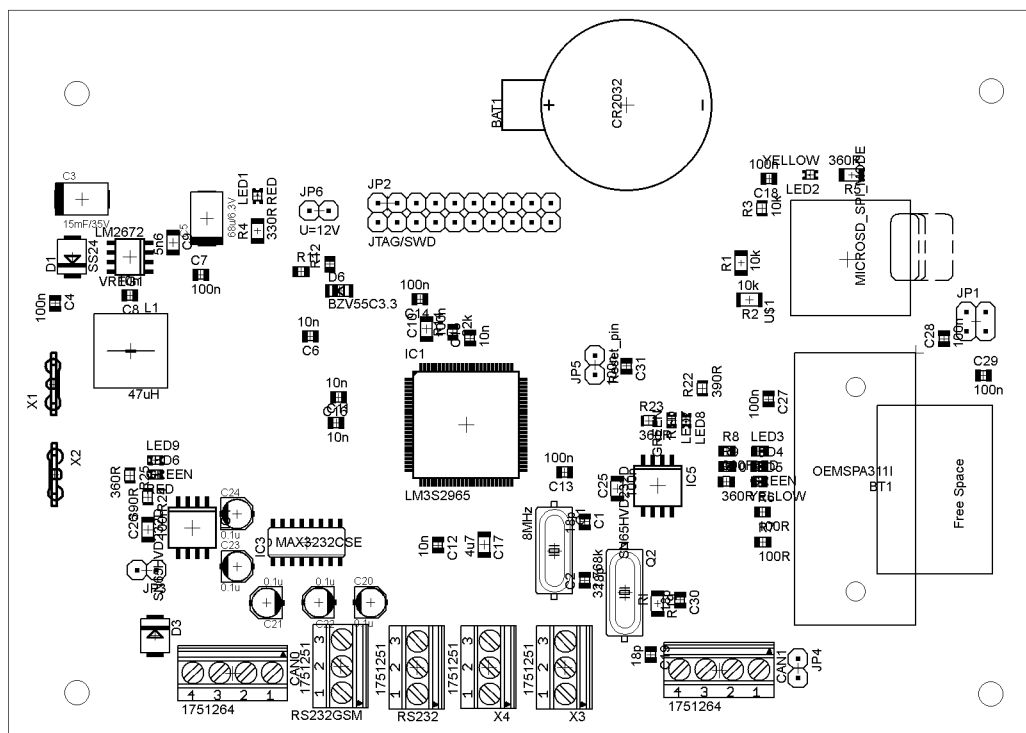
DPS – strana součástek



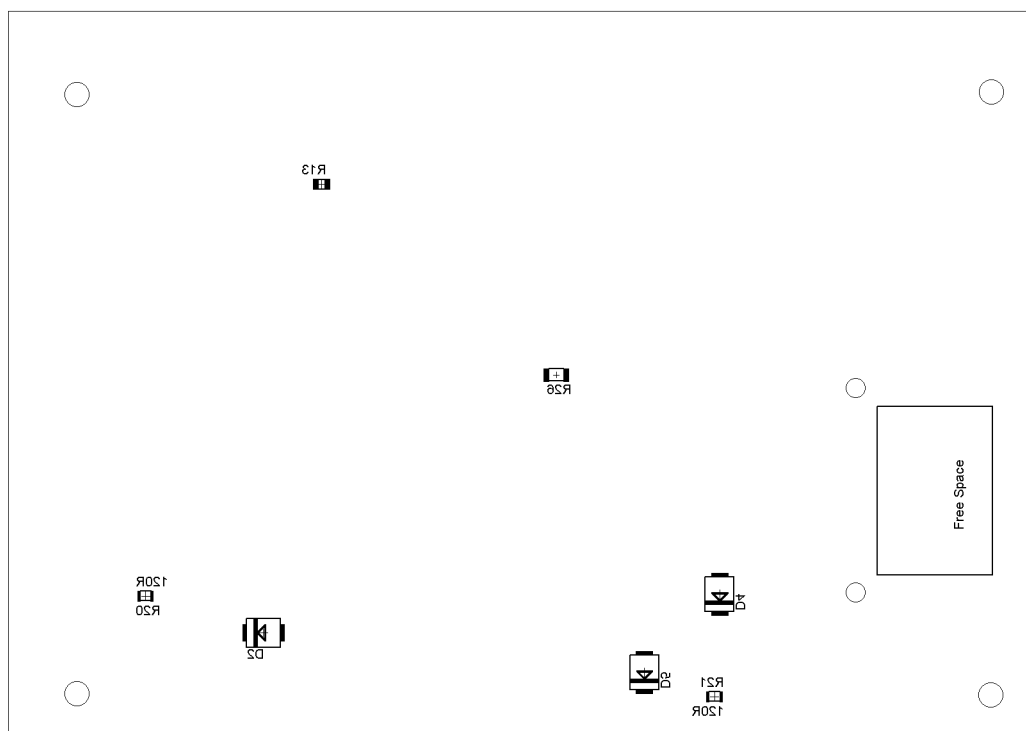
DPS – strana spojů



## DPS – umístění součástek – strana součástek



## DPS – umístění součástek – strana spojů



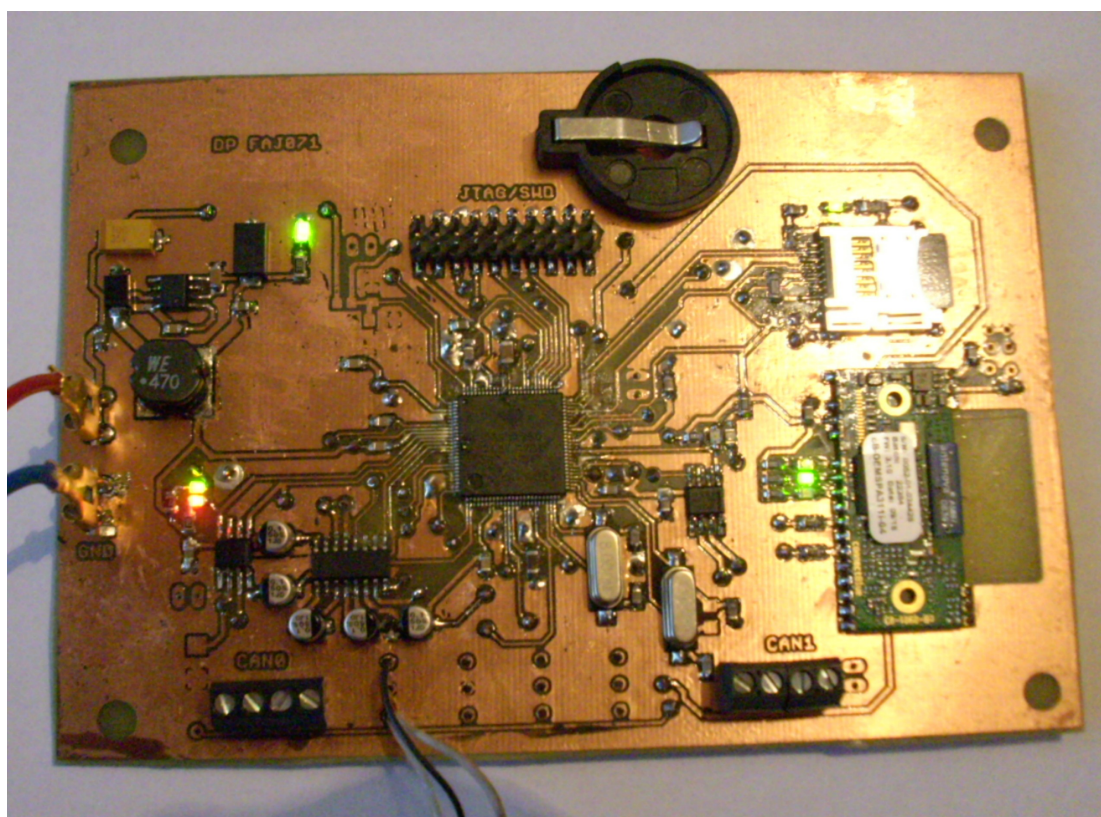
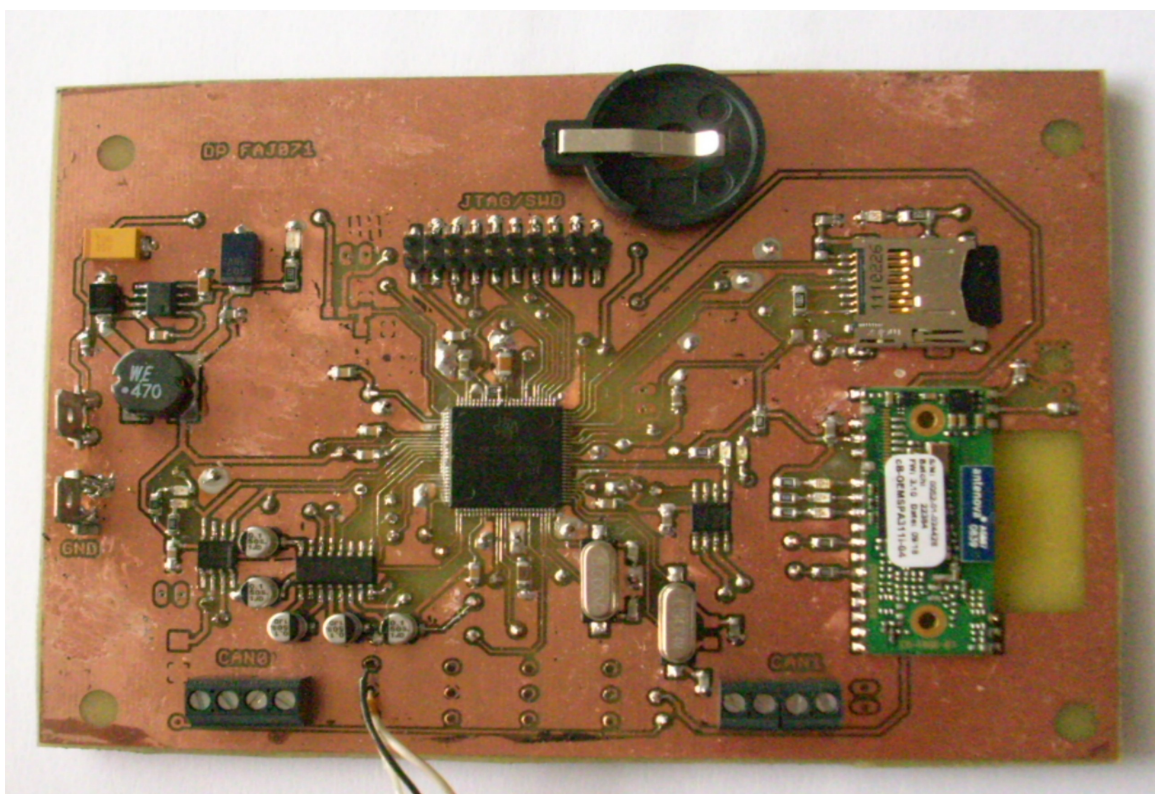
## Seznam součástek

Označení	Hodnota	Pouzdro
BAT1	CR2032	SN2032
BT1	OEMSPA311I	SOLDER
C1	18p	C0805
C2	18p	C0805
C3	15mF/35V	SMC_D
C4	100n	C0805
C5	68u/6.3V	SMC_D
C6	10n	C0805
C7	100n	C0805
C8	10n	C0805
C9	5n6	C0805
C10	100n	C1206
C11	10n	C1206
C12	10n	C0805
C13	100n	C0805
C14	100n	C0805
C15	10n	C0805
C16	10n	C0805
C17	4u7	C0805
C18	100n	C0805
C19	18p	C1206
C20	0.1u	C0805
C21	0.1u	C0805
C22	0.1u	PANASONIC_B
C23	0.1u	PANASONIC_B
C24	0.1u	PANASONIC_B
C25	100n	PANASONIC_B
C26	100n	PANASONIC_B
C27	100n	PANASONIC_B
C28	100n	C1206
C29	100n	C1206
C30	18p	C0805
C31	100n	C0805
CAN0	1751264	C0805
CAN1	1751264	C0805
D1	SS24	C0805
D2	SCHOTTKY-DIODESMD	1751264
D3	SCHOTTKY-DIODESMD	1751264
D4	SCHOTTKY-DIODESMD	SMB
D5	SCHOTTKY-DIODESMD	diode
D6	BZV55C3.3	diode
IC1	LM3S2965	diode
IC3	MAX3232CSE	diode
IC4	SN65HVD232D	diode
IC5	SN65HVD232D	SOD80C
JP1	PINHD-2X2	TQFP100
JP2	JTAG/SWD	SO16
JP3	PINHD-1X2	SOP-08
JP4	PINHD-1X2	SOP-08
JP5	Reset_pin	pinhead
JP6	U=12V	2X10SMD
L1	47uH	pinhead
LED1	RED	pinhead
LED2	YELLOW	1X02
LED3	RED	
LED4	GREEN	

LED5	YELLOW	1X02
LED6	RED	WE-PD2_XL
LED7	GREEN	CHIPLED_0805
LED8	RED	CHIPLED_0805
LED9	GREEN	CHIP-LED0805
Q1	8MHz	CHIP-LED0805
Q2	32,768k	CHIP-LED0805
R1	10k	CHIP-LED0805
R2	10k	CHIPLED_0805
R3	10k	CHIPLED_0805
R4	330R	CHIPLED_0805
R5	360R	CHIPLED_0805
R6	100R	HC49UP
R7	100R	HC49UP
R8	390R	R1206
R9	390R	M1206
R10	360R	M0805
R11	R-EU_R0805	R1206
R12	R-EU_R0805	R1206
R13	R-EU_R0805	R0805
R14	22k	R0805
R19	RI	R0805
R20	120R	R0805
R21	120R	R0805
R22	390R	R0805
R23	360R	rel
R24	390R	rel
R25	360R	rel
R26	1k	R0805
RS232	1751251	R1206
RS232GSM	1751251	M0805
U\$1	MICROSD_SPI_MODE	M0805
VREG1	LM2672	M0805
X1	ST4,8	M0805
X2	ST4,8	M0805
X3	1751251	M0805
X4	1751251	M0805
		R1206
		MICRO_SD_CONNECTOR
		SO08
		con-rib
		con-rib
		1751251
		1751251

## Příloha III

Vyrobená DPS Diagnostického rozhraní elektromobilu





## Příloha IV

### Porovnání GPS modulu s baterií typu AA



## Příloha V

### Připojení GPS modulu k vývojovému kitu CAN

